FLOWGEM: A MULTI-SCALE NEURAL SURROGATE MODEL BASED ON GEOMETRY AND PHYSICS ENRICHMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks learn the low-frequency components of a target signal before the high-frequency ones, a phenomenon known as spectral bias. This failure mode also affects neural surrogate models used to simulate systems governed by PDEs. The bias is particularly consequential in automotive and aerospace aerodynamics, where the flow is shaped both by millimeter-scale geometric details and by meter-scale structures such as the windshield angle. To address this challenge, CFD surrogates commonly employ hierarchical modeling, mixing reads that target local geometry features with reads that target large-scale information, and combining them into a learned latent representation with little control over how these scales interact. In this study, we present FlowGEM, a neural surrogate model trained on surface and volume point clouds. FlowGEM reads the geometry at multiple spatial scales in sequence, with shared weights across scales, recursively conditioning through a scale-dependent gate. In addition, it learns global conditioning through a physics enrichment module, injecting learned physics information into the model's latent geometry representation. With this setup, we compare FlowGEM against state-of-the-art surrogate models on realistic aerodynamics benchmarks (DrivAerML and NASA CRM), where it outperforms them on 4 of the 6 metrics tested. To understand the impact of spectral bias on the model, we analyze the prediction error in spectral bands and show how the multi-scale attention reduces frequency-dependent errors. Finally, because the output is predicted independently at each input point, FlowGEM's inference cost grows linearly with the number of points evaluated, and a full mesh can be decoded in chunks that fit the available memory.

1 INTRODUCTION

Computational fluid dynamics (CFD) plays a central role in many engineering and research disciplines. In the automotive industry, CFD solvers compute solutions of partial differential equations (PDEs) numerically on large-scale three-dimensional meshes. The computational cost of such simulations limits how many design variants can be evaluated within a fixed compute budget, which in turn constrains exploration of the design space for low-drag vehicle geometries. Neural surrogate solvers, which learn to approximate the solutions of the governing PDEs (i.e., the Navier-Stokes equations), have emerged as a promising alternative for rapid evaluation. Recent work has shown that they can predict flow fields with high accuracy at a fraction of the computational cost of conventional solvers.

A central challenge in simulating automotive aerodynamics is the inherent multi-scale character of the problem. The fluid flow is dictated both by global geometric features, such as the inclination of the windshield, and by local features, such as the fine surface details of panel gaps. The physics itself depends on multiple scales: quantities such as wall shear stress are more localized than surface pressure, while integrated quantities such as drag and lift depend on non-trivial interactions between all these scales. In addition, the modeling of the fluid in the volume outside the geometry is particularly sensitive to the multi-scale coupling of the point-to-wall distance and the details of the car surface geometry.

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.

This challenge is exacerbated by the spectral bias of neural networks, a well-documented artifact of training machine learning models. Rahaman et al. (2019) show that neural networks learn low-frequency signals (large-scale structure) first, while the high-frequency signals (local details) are only resolved at later stages of training. Recent work has shown that directly addressing this spectral bias, e.g., with a dedicated loss term, does not necessarily improve the aggregate metrics (Khodakarami et al., 2026b). Instead, current state-of-the-art neural surrogate solvers try to mitigate this challenge either by explicitly including local and global reads (Alkin et al., 2025), by adding hierarchical information through a multi-scale geometry representation (Ranade et al., 2025; Adams et al., 2025), or both. The latter approaches read the geometry itself (e.g., the car surface represented by its mesh or from a point cloud obtained from it) at different scales, either by sampling at different rates or aggregating over different distances, and combine the resulting features into a latent representation of the point where the physics will be estimated (the query point). In general, these strategies collapse the multiple reads into a single representation before the prediction step, usually without cross-talk between scales.

We introduce FlowGEM, a neural surrogate solver built on a natively multi-scale attention operator that reads geometry features recursively, tackling the spectral bias failure mode while maintaining, or improving upon, state-of-the-art performance on public automotive CFD datasets. As a starting point, we adopt the recipe of the AB-UPT family of models, which builds its geometry representation through hierarchical sampling (queries, anchors and substrate). Then, we propose a multi-scale attention module that reads geometry information at each query point in sequential steps, updating the query latent representation conditioned on the scale. We also present a mechanism for global-scale physics enrichment, in which anchor points' geometry representations are updated with physics information read directly from the targets through an auxiliary decoder head and loss term. Finally, we analyze the power spectrum of the model prediction error on DrivAerML to understand its spectral bias.

2 Related Work

Neural operators Neural operators map between function spaces, with DeepONet (Lu et al., 2021) and FNO (Li et al., 2021) being early prominent examples. Extending these ideas to the irregular, large-scale 3D domains typical of engineering geometries requires giving up the regular grid, through graph-based local integration (Li et al., 2023), attentional graph encoders with geometry embeddings (Wen et al., 2025), or gauge-invariant graph embeddings (Rigotti et al., 2026). Designing the operator at the continuous level instead keeps the discretisation error small under changes of mesh density and topology (Zeng et al., 2025). FlowGEM follows this line: it consumes mesh-enriched point clouds and predicts at arbitrary query locations.

Multi-scale geometry encoding DoMINO (Ranade et al., 2025) is a point-cloud neural operator that combines a multi-scale global geometry encoding through convolutions with local stencil-based aggregation. AB-UPT (Alkin et al., 2025) assigns geometry encoding, surface prediction and volume prediction to separate transformer branches that exchange information via cross-attention and uses anchors to reach surface and volume field accuracy on automotive CFD meshes of up to 150 M cells. Transolver-3 (Zhou et al., 2026) optimizes the physics-attention mechanism proposed by Wu et al. (2024) and uses a geometry amortisation training scheme to obtain results for meshes containing up to 10^8 cells. By exploiting hierarchy and multi-scale supervision, HiLNO (Hu et al., 2026b) reduces the parameter count compared to Transolver and LinearNO (Hu et al., 2026a), but its experiments were confined to smaller datasets (Airfoil and Shape-Net Car). Closest to our setting, GeoTransolver (Adams et al., 2025) builds a geometry and boundary-condition context from multi-scale ball queries and re-injects it into every attention block, arguing that conditioning applied only at the input drifts across depth. Across these methods the multi-scale reads are combined by concatenation or pooling into a single representation before prediction, rather than being allowed to condition one another.

Anchors and physics conditioning Compressing a large input set into a small set of latent tokens that queries later read through cross-attention is the Perceiver construction (Jaegle et al., 2021). AB-UPT (Alkin et al., 2025) adapts it to CFD with *anchor* tokens: a sampled subset of points self-attends to share global context, while the remaining queries only read from the anchors. There the

anchors carry geometry, while operating conditions must be supplied as explicit inputs, as in the cross-attention over simulation parameters of SMART (Hagnberger & Niepert, 2026). FlowGEM instead supervises the anchors to carry physics as well, reading the operating conditions from the data rather than having them prescribed.

Spectral bias and neural operators Neural networks learn the low-frequency component of a given target function before learning the high-frequency ones, as characterized by Rahaman et al. (2019). The same bias appears in neural operators, as reviewed by Xu et al. (2025). In these applications, the bias can be detected as over-smoothed predictions lacking sharp details and fine structure of the target fields (Khodakarami et al., 2026b), and it also influences how solutions generalize under shifts in input frequency (Xie & Luo, 2026). Solutions in the literature act at different stages of the model pipeline: Khodakarami et al. (2026a) rescale high-frequency signal components in the latent space of convolutional operators; Khodakarami et al. (2026b) study the loss design, employing the Binned Spectral Loss of Chakraborty et al. (2026) to tackle spectral bias in neural operators for time-dependent applications; Perrone et al. (2026) instead corrects the spectrum after training, using a diffusion posterior sampling conditioned on sparse observations. Closest to the approach taken in this study, Liu et al. (2024) address the bias at the architecture level. Their proposed hierarchical attention operator attends over a hierarchy of levels, each with its own interaction range, and is trained with an H^1 loss that weights high-frequency signal components more heavily. FlowGEM shares the idea of an explicit scale hierarchy, but places it in the per-query geometry read over an unstructured point cloud, rather than in a latent representation of the domain. In addition, our proposal keeps the levels weight-tied and query-independent, so that cost grows linearly with the number of evaluation points and the parameter count stays approximately constant in the number of levels. We quantify the result with a spectral decomposition of the model error in Section 5.2.

3 METHODS

3.1 PROBLEM DESCRIPTION

To build useful pipelines for automotive CFD applications, neural surrogate models must be **scalable**: typical meshes may contain 1–100M cells. Therefore, neither training nor inference may scale prohibitively with the size of the input geometry. In order to save computing time, we achieve this in three steps:

- First, by ignoring mesh connectivity and using a mesh-enriched point cloud as input. This means that the main geometry representation consists of 3D positions on the geometry surface, enriched with mesh-based information such as normal vectors, surface curvature and mesh cell area.
- Second, we perform the training and inference at a fixed sampling budget. This allows the training to scale with surface complexity rather than density, i.e., the model is insensitive to the same geometry (same complexity) being meshed with a different number of cells.
- Third, the field prediction is performed at individual points (queries), which are independent of each other. This requires that at inference time, no query reads information from another query, ruling out strategies such as query self-attention.

Together, these three requirements make model training and inference scale at a controlled rate (sampling choice) based on the geometry complexity, and linearly instead of quadratically with the number of inference points. They also allow for the model to be interpreted as an approximate neural operator: its inputs are points in space (enriched with underlying surface mesh information); however, the discretisation invariance is only approximate, as the sampled point cloud density retains some dependency on the discretisation strategy.

We represent the surface and geometry points through the following: (1) their positions, encoded with RoPE (Su et al., 2021); (2) their normals for surface points, obtained from the underlying mesh; (3) their curvatures, obtained as the minimum and maximum curvature values for the surrounding cells of a given point; (4) their underlying cell area, as a measure of the density in a given geometry location. For volume points, the normals are instead replaced by the normalized vector distance between the point and its surface nearest neighbor, from which the curvature and area are also obtained.

Table 1: Distance from a query to its farthest neighbour at each level, for a test geometry from the DrivAerML dataset. Values are the median over 20k queries, with the 25th and 75th percentiles as sub- and superscripts, in millimetres. Queries and an 80k geometry pool are sampled from the surface mesh.

LEVEL	$b = 2$		$b = 5$	
	STRIDE	SAMPLE	STRIDE	SAMPLE
0	$15.2^{50.9}_{3.5}$	$15.2^{50.9}_{3.5}$	$15.2^{50.9}_{3.5}$	$15.2^{50.9}_{3.5}$
1	$21.4^{68.3}_{4.9}$	$21.4^{69.4}_{5.0}$	$34.2^{101.5}_{7.5}$	$35.0^{105.9}_{7.8}$
2	$30.6^{92.3}_{6.8}$	$30.8^{95.1}_{7.1}$	$76.3^{208.7}_{16.8}$	$85.3^{225.3}_{17.8}$
3	$43.2^{125.1}_{9.5}$	$44.6^{131.6}_{9.9}$	$177.4^{441.3}_{43.7}$	$188.2^{464.5}_{46.6}$

Sampling To represent the geometry, we perform independent sampling at different stages for different purposes. First, we sample query and anchor points. Queries are the positions where the fields will be inferred, while anchors serve to describe large-scale structures and are later enriched with physics information. We also sample a large substrate, the surface geometry pool, which will be used to enrich queries and anchors with raw local shape information in the multi-scale attention step. Anchors are obtained as a subset of the geometry pool, while the queries are required to be disjoint from it. We will refer to the collection of queries, anchors and pool as *points*. This recipe follows closely the sampling strategy used by the AB-UPT model (Alkin et al., 2025).

3.2 GEOMETRY ENRICHMENT THROUGH MULTI-SCALE ATTENTION

The initial point representation is obtained as a linear combination of the geometric features described above. This representation is not detailed enough to describe the physics fields, as it lacks neighboring information. As an initial enrichment strategy, a simple kNN or radius-based gather could be used to pass information from the pool to the queries and anchors. However, to address the multi-scale nature of the problem, we choose to perform this stage in sequential levels.

3.2.1 BUILDING THE LEVELS

The goal of the multi-level approach is to enrich the queries and anchors with point information at multiple distance scales. Our proposed methodology hinges on three hyper-parameters: k, the number of neighbors read at each level; L, the number of enrichment levels; and b, the base of the geometrical level growth. Each level ℓ is then defined as containing the $k \cdot b^\ell$ nearest neighbors, from which k are read. We employ two strategies for performing the k read at each level:

- **Striding**: Each pool point is distance-ranked with respect to each query/anchor. At each level, every b^ℓ-th nearest neighbour is chosen as that level's probes. This strategy involves kNN's over large k's, which can quickly become inefficient.
- **Sampling**: The original pool is sub-sampled into nested subsets. Level ℓ keeps b^ℓ-th point of a single random permutation. Each level then takes k nearest neighbors over a coarser cloud. This strategy depends on ensuring that the sampling reproduces the correct distance distribution profile.

In the table 1, we compare the reach of the different strategies at different levels hyper-parameters. At $b < 5$, we observe that both strategies offer the same performance, both in terms of accuracy and computation time. For $b \geq 5$, performing a kNN pass over O(1000) neighbours becomes consequential, and the sampling strategy is preferred.

3.2.2 SCALE OPERATOR

The multi-scale operator is a Perceiver-like cross-attention module built to enrich the queries and anchors with the reads at each level ℓ. The cross-attention is performed with the position encoded inputs (RoPE) as queries, and input-specific latent representations as keys and values. For a given input point, the neighbors are represented by relational information, this means that instead of encoding raw features, such as positions and normals, they are directly built upon the relation to the

input: position becomes distance to input, the normal vector becomes the angle between the two points' normals. The input-specific representation includes bias terms that encode whether the input point belongs to the surface or volume, and that depend on the size scale of the neighborhood at that level through $\sigma_\ell(i)$, an RMS of all points used to build a level around input point i. This RMS is also used to normalize the input-point distances used to build the KV's. We can write the key-value representation as follows:

$$\text{KV}(i,j)_\ell = \text{Linear}(\text{LayerNorm}(\Delta(i,j))) + W_{\sigma_\ell} \cdot \sigma_\ell(i) + W_{vs}(i) + W_\ell \tag{1}$$

where $\text{KV}(i,j)_\ell$ is the key-value representation of pool point j relative to input point i, $\Delta(i,j)$ are the concatenated relational quantities between input and pool points, W_{σ_ℓ} are bias weights based on the $\sigma_\ell(i)$ RMS, $W_{vs}(i)$ are bias weights based on whether input points are from the surface or volume clouds, and W_ℓ are level-specific bias weights. By defining the KV's representation from relational quantities, it becomes invariant to rotations and translations of the geometry, so the model does not have to learn those invariances from data.

Due to the co-similarity between the levels, the neighborhood reads can be performed by a single cross-attention module that is shared across levels. The challenge, however, is how to control how the different level reads interact towards a final latent representation of the input points. We choose to perform that through feature-wise linear modulation (FiLM) gates (Perez et al., 2018), updating the input representation recursively according to the following equation:

$$h_\ell(i) = \gamma(\sigma_\ell) \cdot S(q = h_{\ell-1}(i), kv = \{\text{KV}_\ell(i,j)\}_j) + \beta(\sigma_\ell) \tag{2}$$

where $\gamma(\sigma_\ell)$ and $\beta(\sigma_\ell)$ are MLPs with only the RBF embedding of the level RMS. Every query and anchor points, across both surface and volume domains, go through the Multi-Scale Attention (MSA) module.

3.3 Physics Enrichment

Geometry-only reading is not enough to provide information about the fluid flow, particularly with regards to operating conditions (e.g., direction of the flow and speed). Instead of specifying those as inputs to the model, we let the anchors read them from the data itself at training time. The anchors which have been geometry-enriched (a_G) need to become sources of physics information, or physics anchors (a_P). As a first step to encode that information, we allow the geometry-enriched anchors to self-attend. This allows for information sharing across surface and volume anchors, building towards a global scale encoding. Then, an anchor-specific decoder head (MLP stack) is applied to each, in order to predict the field values at the anchor locations, separately for each domain. The decoding is controlled by an auxiliary loss, of the same type of the standard loss at query locations. The last intermediate latent at the decoder MLP stack is used to enrich the geometry-enriched anchor representation, then another self-attention layer fully shares the local physics information across the new physics anchors. The information flow can be visualized as follows:

$$\begin{aligned} \text{MSA}(a, \text{pool}) \to a_G, \qquad & \text{Self-Attn}(\{a_G\}) \to a_G^*, \\ a_G^* + w \cdot h_p(a_G^*) \to a_P^*, \qquad & \text{Self-Attn}(\{a_P^*\}) \to a_P, \end{aligned} \tag{3}$$

where w is a learned parameter that controls the physics injection, initialized at 0.02. Both surface and volume anchors are used in this construction, allowing for cross-domain physics to be learned.

3.4 Query-Anchor Read, Decoder and Loss

As stated, one important ingredient in the model is that the queries should not share information across themselves, to avoid query dependence at inference step. Therefore, the anchors are used as the encoders of global physics information. Their interactions is therefore asymmetrical, queries read the anchors through a Perceiver-like cross attention, where the queries are the geometry-enriched point queries, and the key-values are the physics anchors. The surface and volume queries are allowed to read the full set of anchors through independent cross-attention modules.

Table 2: Summary of the datasets used in this study. Sizes are the number of mesh cells per geometry.

DATASET	OBJECTS	CASES	SURFACE	VOLUME	OP. CONDITIONS
DrivAerML	Cars	500	∼8.8M	∼140M	✗
NASA-CRM	Aircraft	149	∼450K	✗	✓

After being enriched both by geometry through the MSA, and by physics through the physics anchors, the queries are ready to be decoded into the physics targets. Each query domain predicts that domain's physics independently: surface queries decode surface-specific targets, volume queries decode volume-specific targets. The decoder head is built through an MLP stack with side-connections, allowing for interdependence between the physics of a given query.

4 EXPERIMENTS

4.1 DATASETS

The model is trained and evaluated on two public benchmark datasets of realistic aerodynamics, summarised in table 2 and described in details in Appendix A.2. DrivAerML (Ashton et al., 2025) provides hybrid RANS-LES simulations of 500 variants of the DrivAer car geometry, varied parametrically, with both surface and volume fields. The NASA Common Research Model (NASA CRM) (Bekemeyer et al., 2025) contains surface data with varied operating conditions, providing a different challenge than DrivAerML.

4.2 ARCHITECTURE

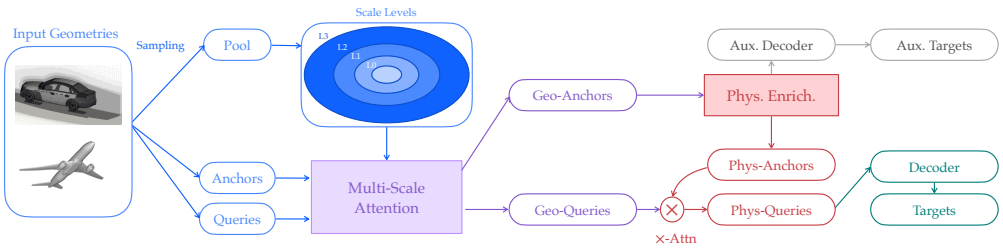

Figure 1: FlowGEM architecture schema showing its three main components: geometry sampling, multi-scale attention and physics enrichment.

The main architecture used in this study uses a hidden dimension of 320 and 10 heads. The MSA is setup with 4 levels, reading 8 nearest neighbors per level at base 2 (details on the physical reach of that configuration are described in table 1). The architecture schema is presented in Figure 1, showing the main modeling steps. The training is performed with a random sample of queries, anchors and surface pool at every forward pass, for 500 epochs. We opted for the Lion optimizer, which has been found to be less sensitive to large sample-by-sample variations due to its independence to the gradient norm, configured with a target learning rate of $5e^{-5}$ scaled by the number of GPUs used in the training, and a weight decay of 0.05. Lion is coupled with a cosine scheduler with 25 warm-up epochs. MSE is used as loss. The same hyper-parameters are used for both datasets. The training is performed at bf16 precision, with 4 H100 GPUs with distributed data parallelism (DDP), training at around 40 seconds per epoch. The final model contains approximately 28M parameters and is fully described in Appendix B. For the sampling steps, we choose a surface geometry pool of 80k random points, 20k anchors and 20k queries for both surface and volume. To stabilize the impact of random draws over the training forward passes, we employ exponential moving average (EMA) weight averaging at the evaluation step.

Table 3: Comparison of different models on the NASA-CRM and DrivAerML datasets. Relative L^2 numbers are presented in percentage. Metrics over multi-component quantities are calculated at vector level.

MODELS	NASA-CRM		DRIVAERML			
	p_s	C_f	p_s	$\boldsymbol{u}$	$\boldsymbol{\tau}$	p_v
AB-UPT	9.77	6.43	3.82	5.93	7.29	6.08
Transolver	9.61	7.04	4.81	6.78	8.95	7.74
Transolver++	9.51	6.95	4.12	<u>4.70</u>	6.42	6.70
TRANSOLVER-3	<u>8.71</u>	<u>5.85</u>	<u>3.71</u>	**4.14**	**5.85**	<u>5.72</u>
FLOWGEM	**6.26**	**4.38**	**3.34**	4.75	<u>6.35</u>	**5.00**

4.3 EVALUATION METRICS

Evaluation is performed at full-mesh: all points in both surface and volume meshes are evaluated, an averaged is obtained per geometry and the average of the per-geometry metrics are used. Due to the query independence, we can chunk the evaluation procedure into independent batches to obtain the final result, which then becomes bottleneck by parallelization. The main result metric is the relative L2 error, as defined in Appendix A.1.1, which is directly comparable to the reference literature. For the integrated coefficients calculation, agreement is quantified via R^2, defined in Appendix A.1.2.

5 RESULTS

The computing costs for training FlowGEM are dataset dependent. Here, we report those representative of the DrivAerML dataset, a realistic automotive case with roughly 8M surface points and 140M volume points. DrivAerML training for 500 epochs costs 33 GPU-hours. At inference time, cost is dominated by the number of queries, growing linearly due to the query-independence requirement. We measure 13.5 ms and 0.05 GB of peak GPU memory per thousand queries on a single GPU. A full surface mesh inference can therefore be performed in chunk sizes as allowed by the available hardware, with no intermediate state caching. The scaling with anchors is less favorable, due to the multiple self-attention steps in the Physics Enrichment module. While modern attention kernels such as flash-attention (Dao et al., 2022) allow for less than quadratically scaling, the growth with the number of anchors is still above linear.

5.1 AGGREGATE METRICS

We present the results with the aggregated metrics over the investigated datasets. To ensure the numbers are comparable, we have reproduced the AB-UPT results locally for the DrivAerML dataset under our own training and evaluation pipelines. Our results agree with the numbers stated in the AB-UPT and Transolver 3 papers. Therefore, we report the remaining baselines as published in the latter, restricting our comparisons to the top three highest accuracy models therein.

The full results comparison is displayed in table 3, showing FlowGEM ahead of previous models in 4 out of 6 metrics investigated . For the NASA-CRM metrics, FlowGEM is ahead of all previous baselines by 25–28%. For DrivAerML, FlowGEM stays ahead in the pressure targets (both surface and volume) while trailing only Transolver 3 on wall shear stress and Transolver 3 and Transolver++ on volume fluid velocity. It's notable that FlowGEM is more performant than AB-UPT, which also employs a sampling based strategy to achieve scalability and query independence.

One clear advantage of full-mesh models such as Transolver 3 is the physics conditioning over the entire mesh. The Transolver family physics attention projects all the mesh cells into M slices, and every cell contributes to the final read at inference time. Therefore, a point's prediction is conditioned over all points jointly, requiring amortized training on subsets and a cached state to decode the full mesh. The two quantities where Transolver 3 leads (wall shear stress and volume velocity, relative to pressure) are the most locally determined. We hypothesise that the this lead is due to the denser mesh available at conditioning (the entire mesh, for Transolver 3). FlowGEM reads a fixed pool of 80k, which bounds the finest scale queries can resolve. We note, however, that

the Transolver 3 official implementation (THUML, 2026) calculates relative L^2 metrics over multi-component targets (surface wall shear stress and volume velocity) using their magnitude. Instead, we report the joint metric in equation 4, a recipe described in both the AB-UPT (Alkin et al., 2025) and Transolver 3 (Zhou et al., 2026) papers. Using the magnitude instead, FlowGEM's DrivAerML relative L^2 errors are: **5.38%** for $|\boldsymbol{\tau}|$ and **3.75%** for $|\boldsymbol{u}|$, below the corresponding Transolver 3 values.

In addition to the field metrics, we evaluate the prediction of integrated quantities, in particular the drag coefficient C_d and the lift coefficient C_l. The drag coefficient is the dimensionless quantity proportional to the ratio between the drag force (the force component in the direction of the flow velocity) and the kinetic pressure times. The lift coefficient performs a similar ratio but with the lift force instead (force component perpendicular to the flow velocity). These are of particular interest for engineering applications, as they help quantify the aerodynamic forces generated by an object moving through a fluid.

We present the predicted versus target C_d and C_l for the test DrivAerML geometries in Figure 2. The coefficients are obtained by integrating the predicted surface pressure and wall shear stress components over the full mesh, normalised by the constant baseline reference area provided by the DrivAerML dataset. FlowGEM predicts C_d with $R^2 = 0.986$ and C_l with $R^2 = 0.979$, versus $0.963/0.975$ for AB-UPT and $0.972/0.985$ for Transolver 3. The drag coefficient is predicted more accurately than by either baseline, while trailing only Transolver 3 on the lift coefficient.

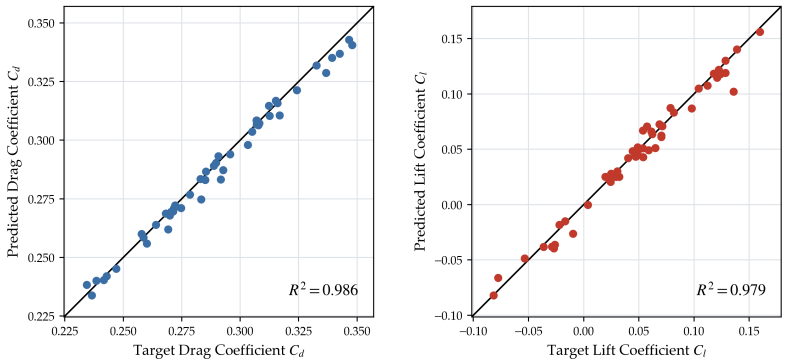

Figure 2: Predicted versus target integrated coefficients on the DrivAerML test geometries. Left: drag coefficient C_d, right: lift coefficient C_l. The diagonal lines represent the perfect agreement.

5.2 Spectral Analysis

In CFD spectral domain, low frequencies corresponds to smooth and large-scale spatial variations such as broad flow trends or global pressure distribution, while high frequencies define sharp, oscillatory, fine-scale spatial variations such as shear layers, wake boundaries and separation lines. Intermediate frequencies capture mid-scale coherent structures of the time-averaged flow, such as pillar vortices and the extent of the wake. Neural networks suffer form know spectral bias where neural networks first learn low-frequency components and under-fit high-frequency components Rahaman et al. (2019).

In order to assess how well our approach capture different scales we analyze the the prediction error in the graph spectral domain on DrivAerML dataset. We build a graph from surface mesh and a k-NN graph over volume points and use its normalized Laplacian L. A full eigen-decomposition is not feasible at this scale. For this reason, we used Chebyshev polynomial filters Sen & Maulik (2026), which approximate graph-spectral frequency bands on unstructured meshes by replacing expensive $O(N^3)$ full eigen decompositions with sparse matrix polynomials. Chebyshev filter-bank approximation bypasses explicit eigenvector computation. Each field is filtered with sparse matrix polynomials to approximate triangular (Barlett) spectral windows with polynomial coefficients obtained by Chebyshev quadrature. In our experiments, we adopt their configuration of $K = 16$,

$M = 32$ bands and $Q = 256$ quadrature points. Adjacent windows overlap by construction, so the bands should be read as a smooth sampling of the spectrum.

In Figure 3, we see the relative error spectral energy of the surface pressure and volume total pressure coefficient, for the AB-UPT model, FlowGEM and a variant of FlowGEM with only one local enrichment step instead of the 4-levels Multi-Scale Attention used in the aggregate results section (FlowGEM-l1). The calculation is performed over 5 random test meshes (the same for all variants), and the median and IQR are plotted. In these plots, flat lines would indicate models that distribute their errors evenly across scales. Models show a high concentration of errors at the same part of the spectrum, around $\lambda = 1$, a fact that is hidden by the usual aggregated metrics. As expected, FlowGEM's error is lower than AB-UPT's particularly in the spectral region where the errors concentrate, while for very low and very high frequencies the errors are at the same level. Comparing FlowGEM $L = 4$ versus $L = 1$, we see that the benefit is not a uniform reduction of the error, instead, on the surface pressure, the biggest gain falls exactly in the mid-range frequency with around 14% error reduction and falling to 5% in the extremes. Both configurations are indistinguishable for the volume total pressure coefficient, which indicates that the multi-scale read, at its current implementation, acts more optimally on the surface geometry rather than on the volume field.

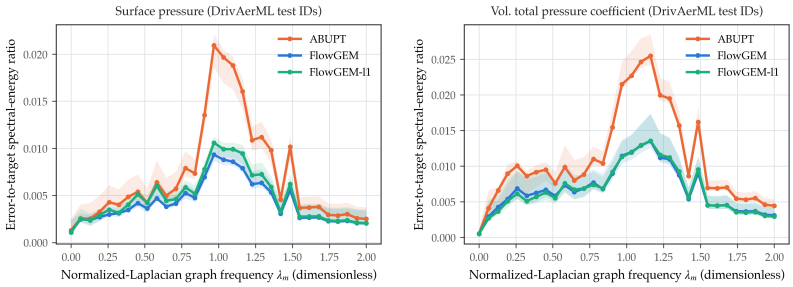

Figure 3: Spectral power decomposition of the target error relative to the spectral power decomposition of the target signal. Left: on the surface pressure, right: on the volume total pressure coefficient.

6 Conclusion

This paper presents FlowGEM, a new neural surrogate model built to solve three main problems: scalability, scale dependency and physics conditioning. With it, we address scalability with a query-independence requirement, the scale dependency with the Multi-Scale Attention module, and the physics conditioning with the Physics Enrichment module. FlowGEM achieves state-of-the-art in 4 of the 6 aggregated metrics in which it was evaluated and compared to other recent, similar models, in automotive and aerospatial realistic datasets. In particular, we demonstrate that the Multi-Scale Attention strategy does mitigate the frequency dependence of the prediction error, particularly for surface targets. These results are obtained while retaining the query independence and linear inference scaling of other anchored, sampling-based models such as AB-UPT, and matching or exceeding the accuracy of Transolver-3, which requires amortized training on subsets of data and cached states to decode the full mesh.

Limitations and Future Work FlowGEM trails Transolver-3 on wall shear stress and volume velocity, the two targets expected to be most sensitive to local signal. We expect a dependency on the density of the geometry pool, which dictates the finest scales to which the model has access. However, scaling it further is currently limited by training memory, as activations grow with the number of sampled points. Reducing that footprint is therefore a requirement for properly testing that specific scaling law. The benefit of the multi-level MSA is clear on the surface pressure, as seen in the spectral analysis, but absent in the volume quantities. This suggests that the current level building and conditioning are better suited for surface scales than volume. Finally, the levels are combined recursively with each read conditioned on the previous. Allowing the scales to interact directly rather than through a single running state is a natural next development.

AI DISCLOSURE

We used generative AI tools (Claude, Codex) to assist and review software implementation of methods, design and results of experiments, and better structure the paper content. We did not use generative AI to generate synthetic data or develop theoretical models behind the papers assumptions. All AI-assisted work was reviewed by the authors.

REPRODUCIBILITY

The datasets used in this study, DrivAerML (Ashton et al., 2025) and NASA-CRM (Bekemeyer et al., 2025), are open datasets publicly available. For DrivAerML, the train/test/validation splits are similar to those used by Alkin et al. (2025) and Zhou et al. (2026), while for NASA-CRM, the splits are dataset pre-defined. AB-UPT results were reproduced in our own pipeline and agree with the published values, while the remaining baselines are quoted as published. A FlowGEM inference implementation is intended for public release, subject to on-going internal approval.

REFERENCES

Corey Adams, Rishikesh Ranade, Ram Cherukuri, and Sanjay Choudhry. GeoTransolver: Learning Physics on Irregumar Domains Using Multi-scale Geometry Aware Physics Attention Transformer, December 2025. URL http://arxiv.org/abs/2512.20399. arXiv:2512.20399 [cs.LG] version: 1.

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. ISSN 2835-8856. URL https://openreview.net/forum?id=nwQ8nitlTZ.

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, April 2025. URL http://arxiv.org/abs/2408.11969. arXiv:2408.11969 [physics.flu-dyn].

Philipp Bekemeyer, Nathan S. Hariharan, Andrew M. Wissink, and Jason Cornelius. Introduction of applied aerodynamics surrogate modeling benchmark cases. *AIAA SCITECH 2025 Forum*, 2025. URL https://api.semanticscholar.org/CorpusID:276328877.

Dibyajyoti Chakraborty, Arvind T. Mohan, and Romit Maulik. Binned spectral power loss for improved prediction of chaotic systems, 2026. URL https://arxiv.org/abs/2502.00472.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

Jan Hagnberger and Mathias Niepert. SMART: Scalable Mesh-free Aerodynamic Simulations from Raw Geometries using a Transformer-based Surrogate Model, 2026. URL https://arxiv.org/abs/2601.18707.

Wenjie Hu, Sidun Liu, Peng Qiao, Zhenglun Sun, and Yong Dou. Transolver is a linear transformer: Revisiting physics-attention through the lens of linear attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 408–416, 2026a.

Zhicheng Hu, Jiacheng Li, and Min Yang. HiLNO: A Hierarchical Latent Neural Operator with Multi-Scale Supervision for PDEs on General Geometries, 2026b. URL https://arxiv.org/abs/2609.18419.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General Perception with Iterative Attention. In *International Conference on Machine Learning*, pp. 4651–4664, 2021.

Siavash Khodakarami, Vivek Oommen, Aniruddha Bora, and George Em Karniadakis. Mitigating spectral bias in neural operators via high-frequency scaling for physical systems. *Neural Networks*, 193:108027, January 2026a. ISSN 08936080. doi: 10.1016/j.neunet.2025.108027. URL https://linkinghub.elsevier.com/retrieve/pii/S0893608025009074.

Siavash Khodakarami, Vivek Oommen, Nazanin Ahmadi Daryakenari, Maxim Beekenkamp, and George Em Karniadakis. Spectral bias in physics-informed and operator learning: Analysis and mitigation guidelines, 2026b. URL https://arxiv.org/abs/2602.19265.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, 2021.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-Informed Neural Operator for Large-Scale 3D PDEs. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 35836–35854. Curran Associates, Inc., 2023. doi: 10.52202/075280-1556. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/70518ea42831f02afc3a2828993935ad-Paper-Conference.pdf.

Xinliang Liu, Bo Xu, Shuhao Cao, and Lei Zhang. Mitigating spectral bias for the multiscale operator learning, 2024. URL https://arxiv.org/abs/2210.10890.

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.

OpenFOAM. OpenFOAM v2212, 2022. https://www.openfoam.com, ESI/OpenCFD Ltd., accessed 2026-09-25.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. FiLM: Visual Reasoning with a General Conditioning Layer . In *AAAI*, 2018.

Niccolò Perrone, Fanny Lehmann, Stefania Fresca, and Filippo Gatti. Correcting neural operator spectral bias via diffusion posterior sampling with sparse observations, 2026. URL https://arxiv.org/abs/2606.03936.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 5301–5310. PMLR, May 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html.

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, January 2025. URL http://arxiv.org/abs/2501.13350. arXiv:2501.13350 [cs.LG].

Mattia Rigotti, Nicholas Thumiger, and Thomas Frick. GIST: Gauge-Invariant Spectral Transformers for Scalable Graph Neural Operators, May 2026. URL http://arxiv.org/abs/2603.16849. arXiv:2603.16849 [cs.LG].

Kanad Sen and Romit Maulik. Scale-aware learning of chaotic dynamics on unstructured meshes via binned spectral losses, 2026. URL https://arxiv.org/abs/2607.19387.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, 2021.

THUML. Transolver-3 (official implementation). https://github.com/thuml/Transolver-3, 2026. Commit ef4fee9, accessed 2026-09-26.

John C. Vassberg, Mark A. DeHaan, S. Melissa Rivers, and Richard A. Wahls. Development of a Common Research Model for Applied CFD Validation Studies. In *26th AIAA Applied Aerodynamics Conference*, Honolulu, HI, 2008. doi: 10.2514/6.2008-6919.

Shizheng Wen, Arsh Kumbhat, Levi 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. 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*, 2024.

Runlong Xie and An Luo. Frequency bias and ood generalization in neural operators under a variable-coefficient wave equation, 2026. URL https://arxiv.org/abs/2605.12997.

Zhi-Qin John Xu, Lulu Zhang, and Wei Cai. On understanding and overcoming spectral biases of deep neural network learning methods for solving PDEs. *Journal of Computational Physics*, 530:113905, June 2025. ISSN 00219991. doi: 10.1016/j.jcp.2025.113905. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999125001883.

Chenyu Zeng, Yanshu Zhang, Jiayi Zhou, Yuhan Wang, Zilin Wang, Yuhao Liu, Lei Wu, and Daniel Zhengyu Huang. Point Cloud Neural Operator for Parametric PDEs on Complex and Variable Geometries, June 2025. URL http://arxiv.org/abs/2501.14475. arXiv:2501.14475 [math.NA].

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, June 2026. URL https://arxiv.org/abs/2602.04940.

A Appendix

A.1 Evaluation Metrics

A.1.1 Mean Relative L^2 Error

For a predicted field $\hat{Y} \in \mathbb{R}^{N \times d_{\text{out}}}$, and the ground truth $Y \in \mathbb{R}^{N \times d_{\text{out}}}$, with N as the number of points, d_{out} is the number of dimensions of a particular target, and N_s the number of training samples (geometries) we have:

$$\hat{L}^2_{\text{rel}}(Y, \hat{Y}) = \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{\|\hat{Y}_s - Y_s\|_2}{\|Y_s\|_2} = \frac{1}{N_s} \sum_{s=1}^{N_s} \frac{\sqrt{\sum_{n=1}^{N} \sum_{d=1}^{d_{\text{out}}} (\hat{y}_{s,n,d} - y_{s,n,d})^2}}{\sqrt{\sum_{n=1}^{N} \sum_{d=1}^{d_{\text{out}}} y_{s,n,d}^2}} \tag{4}$$

A.1.2 R^2

For the integrated coefficients, we report the coefficient determination of the N_s test geometries:

$$R^2 = 1 - \frac{\sum_{s=1}^{N_s} (\hat{c}_s - c_s)^2}{\sum_{s=1}^{N_s} (c_s - \bar{c})^2} \tag{5}$$

where c_s and $\hat{c}_s$ are the reference and predicted coefficients of geometry s and $\bar{c}$ is the mean reference value. Note that R^2 measures error relative to the variance of the test set, so a wider spread of reference values yields a higher R^2 for the same prediction error.

A.2 DATASETS

DrivAerML The DrivAerML by Ashton et al. (2025) dataset was created with the aim to improve on the scarcity of full CFD simulations of realistic vehicles for the automotive industry. It contains 500 parametrically constructed variants of the DrivAer reference vehicle, with an OpenFOAM (OpenFOAM, 2022) physics simulation running with highly accurate hybrid RANS-LES methods. Each geometry variant contains approximately 8 million surface cells and 140 million volume cells, at the same operating conditions. Following common benchmarks using this dataset, we focus on the pressure $\boldsymbol{p_s}$ and the 3D wall shear stress $\boldsymbol{\tau}$ as surface targets, and the total pressure coefficient $\boldsymbol{p_v}$ and 3D fluid velocity for the volume $\boldsymbol{u}$. Of the 500 geometries provided, 400 are used for training, 50 for testing and 34 for validation.

NASA CRM The NASA Common Research Model (CRM) by Vassberg et al. (2008) is an open source aircraft dataset, containing CFD data on full body aircraft geometries, simulated with the DLR TAU RANS solver and the Spalart–Allmaras turbulence model. It contains surface-only data on 149 variants, with a pre-defined train/test split, with about 450k mesh points. One important distinction with respect to DrivAerml is the varying operating conditions at each sample. Following the recipe in Zhou et al. (2026), we utilize six extra input parameters to overcome this dependency: Mach number, angle of attack, and control surface deflections. The operating parameters are combined with an MLP and appended to the anchor representation before the physics enrichment. The physics targets regressed are the surface pressure $\boldsymbol{p_s}$ and the friction coefficient $\boldsymbol{C_f}$.

B MODEL AND TRAINING HYPERPARAMETERS

Table 4 lists the configuration used for every result in this paper. The configuration is mostly used for both datasets, except the data-specific paths: NASA-CRM carries no volume information, and its six operating parameters are supplied as additional inputs.

Table 4: FlowGEM hyperparameters.

Architecture	
Hidden dimension	320
Attention heads	10
MLP ratio	4.0
Parameters	28.7M
Multi-scale attention	
Levels L	4
Neighbours per level k	8
Level growth base b	2
Cross-attention scale blocks	2
Level conditioning	FiLM on σ_ℓ
Physics enrichment	
Geometry self-attention blocks	4
Injection weight (init.)	0.02
Physics self-attention blocks	1
Sampling	
Queries per read	20k
Anchors	20k
Geometry pool	80k
Decoder	
MLP blocks with side connections	4
Optimisation	
Optimiser	Lion
Peak learning rate	2×10^{-4}
Weight decay	0.05
Schedule	cosine, 25 warm-up epochs
Epochs	500
Batch size (per GPU)	1
Loss	MSE, equal weights
Precision	bf16
Hardware	4 $\times$ NVIDIA H100