FCMO: A Flow-Curv Mamba Operator for Large-Scale 3D Vehicle Aerodynamics

Yuchen Xie[1*], Yufeng Xie[2*], Hanyu He[1], Yue Huang[3], Lijuan Sun[3], Henyi Ren[1†]

[1]College of Information Science and Technology & Artificial Intelligence, Nanjing Forestry University, Nanjing, China
[2]College of Mechanical and Electronic Engineering, Nanjing Forestry University, Nanjing, China
[3]College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China
{xieyuchen, xyjj, 2211501208, renhy}@njfu.edu.cn, {huangyue, sunlijuan}@njupt.edu.cn

Abstract

Large-scale three dimensional vehicle aerodynamics prediction poses critical computational challenges in modern automotive design, where traditional CFD methods require prohibitive simulation times that conflict with rapid design iteration demands. While recent neural operator approaches show promise, existing methods struggle with computational complexity in dense meshes and fail to preserve essential topological information when processing large-scale point clouds. We propose FCMO, a physics-aware neural operator that integrates fluid mechanics principles with selective state space modeling for efficient large-scale vehicle aerodynamics. FCMO introduces four synergistic components: FlowCurv Anchor Sampling that intelligently selects mesh nodes based on normalized local curvature and windward sensitivity. Additionally, dual-scale physics-aware position encoding with adaptive k-NN construction transforms 3D irregular meshes into causality-preserving sequences through feature-guided serpentine scanning. The model integrates a flow-aware Mamba processor incorporating selective mechanisms that dynamically modulate state transitions based on wall distance and flow characteristics. Finally, a physics-constrained decoder enforces conservation laws through mixed weighted interpolation. Extensive experiments on Ahmed-Body and DrivAerNet benchmarks demonstrate that FCMO achieves consistent state-of-the-art performance with 5.2% improvement in surface pressure prediction, 9.3% enhancement in wall shear stress estimation, and 11.4% boost in drag coefficient accuracy, while maintaining superior computational efficiency with 9.4% fewer FLOPs and 9.9% reduced memory usage compared to existing methods.

Introduction

Aerodynamic prediction of large-scale three-dimensional vehicle geometries represents one of the most critical challenges in the modern automotive industry (Ahmed, Ramm, and Faltin 1984). Accurate prediction of vehicle surface pressure distribution and drag coefficients directly determines vehicle performance optimization and design innovation effectiveness. Traditional computational fluid dynamics (CFD) methods solve the Navier-Stokes equations to predict vehicle surface pressure distribution (Lee and Moser 2015; Dhaubhadel 1996), yet computational times ranging from hours to days fundamentally contradict the urgent demands for rapid iterative optimization in modern automotive design (Kochkov et al. 2021).

To overcome the computational bottlenecks of traditional CFD methods, machine learning techniques have demonstrated tremendous potential for aerodynamic simulations (Kovachki et al. 2023). Neural operator methods achieve remarkable generalization capabilities by learning mappings between infinite-dimensional function spaces. Fourier Neural Operator (FNO) (Li et al. 2020) learns operator mappings in the frequency domain through Fourier transforms, while Geometry-Informed Neural Operator (GINO) (Li et al. 2023) effectively handles irregular geometric domains by combining graph neural operators with Fourier architectures. Graph Neural Network (GNN) methods have also shown outstanding performance, with MeshGraphNet (Pfaff et al. 2020) learning mesh-based simulations through message passing mechanisms. Recent Transformer architectures have gained attention for global modeling capabilities, where GNOT (Hao et al. 2023) uses heterogeneous normalized attention for multi-input functions and irregular meshes, while Transolver (Wu et al. 2024) captures high-order correlations through physics-aware attention. The recent AeroGTO (Liu et al. 2025) combines local feature extraction with global correlation capturing, specifically designing an efficient graph-transformer operator for large-scale aerodynamics.

However, existing methods face critical challenges when handling large-scale automotive aerodynamics. Traditional GNNs exhibit computational complexity proportional to the number of edges in dense meshes, manifesting as approximately quadratic growth in large-scale three-dimensional automotive meshes (Xu et al. 2018; Cao et al. 2023). The oversmoothing problem induced by multi-layer message passing suppresses high-frequency signal propagation (Rusch, Bronstein, and Mishra 2023). While Transformer methods possess global modeling capabilities, they are prone to losing critical topological information when processing large-scale point clouds, limiting accurate capture of complex aerodynamic correlations (Liu et al. 2025).

State Space Models (SSMs), particularly the Mamba ar-

*These authors contributed equally.
†Corresponding author.

chitecture, have demonstrated remarkable success in image and natural language processing through effective sequentialization strategies (Gu and Dao 2023; Dao and Gu 2024). Mamba's selective state space mechanism offers significant potential for CFD applications through input-dependent parameter adaptation, enabling dynamic processing of spatially-varying characteristics fundamental to flow analysis (Gu and Dao 2023). Furthermore, Mamba's linear sequence length complexity and enhanced state capacity provide essential computational advantages for large-scale CFD tasks (Gu and Dao 2023; Dao and Gu 2024). Recent advances in Mamba-based neural operators demonstrate substantial acceleration in PDE solving tasks while maintaining accuracy (Liu et al. 2024b). However, existing sequentialization approaches designed for regular data structures cannot be directly applied to irregular 3D automotive meshes, where complex geometric topologies and varying node densities present fundamental challenges for sequential processing.

To address these computational complexity and topological preservation challenges in large-scale vehicle aerodynamics, we introduce FCMO, a physics-aware neural operator that integrates fluid dynamics principles into Mamba's selective state space modeling. Given a 3D mesh of car geometry, FCMO accurately predicts surface pressure distributions and estimates drag coefficients. FCMO incorporates four synergistic components, including a FlowCurv sampling strategy that intelligently selects aerodynamically critical regions, a Flow-Aware Mamba processor that modulates state transitions based on wall distance and flow characteristics, dual-scale physics-aware position encoding with adaptive k-NN construction that transforms irregular meshes into causality-preserving sequences via feature-guided serpentine scanning, as well as a physics-constrained decoder that enforces conservation laws through mixed weighted interpolation. This design effectively captures long-range flow dependencies while maintaining linear computational complexity, enabling rapid and accurate inference with enhanced physical consistency across complex datasets.

Overall, the main contributions of this work include:

- We introduce the first specialized Mamba neural operator designed for computational fluid dynamics tasks, addressing fundamental challenges in large-scale automotive aerodynamics. Our framework provides a physics-aware solution that integrates fluid mechanics principles into selective state space modeling, overcoming quadratic computational complexity and topological information loss in existing methods.
- We develop core technical innovations including a physics-informed sampling strategy combining local curvature and windward sensitivity, a physics-aware adaptation of selective state space models incorporating critical physical features, and a causality-preserving architecture transforming irregular geometries into flow-directional sequences while achieving linear computational complexity.
- We conduct extensive experiments on Ahmed-Body and DrivAerNet benchmarks, achieving consistent state-of-the-art performance with substantial improvements in surface pressure prediction, wall shear stress estimation, and drag coefficient accuracy, while delivering significant computational advantages with reduced operations and memory usage compared to existing approaches.

Related Work

Neural Operators

Neural operators have revolutionized PDE solving by learning mappings between infinite-dimensional function spaces (Kovachki et al. 2023). The foundational Fourier Neural Operator (FNO) (Li et al. 2020) achieves discretization-invariant learning through spectral domain parameterization, while GINO (Li et al. 2023) addresses irregular geometries by combining graph neural operators with FNO architectures. Recent advances include Physics-Informed Neural Operators (PINO) (Wang, Wang, and Perdikaris 2021), which eliminate training data requirements by incorporating PDE constraints directly into the loss function. Zhang et al. (Zhang et al. 2024) propose blending neural operators with relaxation methods in PDE numerical solvers for enhanced computational efficiency, while O'Leary-Roseberry et al. (O'Leary-Roseberry, Chen, and Ghattas 2024) introduce derivative-informed neural operators for high-dimensional parametric derivative learning. Liu et al. (Liu et al. 2024a) develop latent neural operators that reduce computational costs by performing operator learning in compressed latent spaces for solving forward and inverse PDE problems. Recent work on Mamba Neural Operator (Liu et al. 2024b) demonstrates that selective state space models can capture long-range dependencies and continuous dynamics more effectively than traditional Transformers, establishing theoretical connections between SSMs and neural operators for enhanced PDE modeling capabilities. Additionally, Transolver (Wu et al. 2024) proposes physics-aware attention mechanisms that decompose domains into learnable slices, achieving linear complexity through projection-inspired attention.

Geometric Deep Learning

Various methodologies have emerged for processing complex geometrical structures, collectively termed geometric deep learning (Bronstein et al. 2017). Graph neural networks are prominent solutions utilizing graph-based kernels for learning representations (Scarselli et al. 2009). For scattered point data, PointNet (Qi et al. 2017a) and PointNet++(Qi et al. 2017b) have been introduced, while MeshGraphNet(Pfaff et al. 2020) leverages sequential message-passing mechanisms for mesh-based simulation domains. However, conventional graph-based approaches encounter computational bottlenecks with quadratic complexity growth relative to mesh resolution (Xu et al. 2018; Cao et al. 2023) and suffer from oversmoothing effects that diminish high-frequency signals (Rusch, Bronstein, and Mishra 2023). Recent developments address these challenges through mesh-based GNN surrogates (Gladstone et al. 2024) and multifidelity graph neural networks (Taghizadeh, Karpatne, and Luo 2025) incorporating auxiliary connections. AeroGTO (Liu

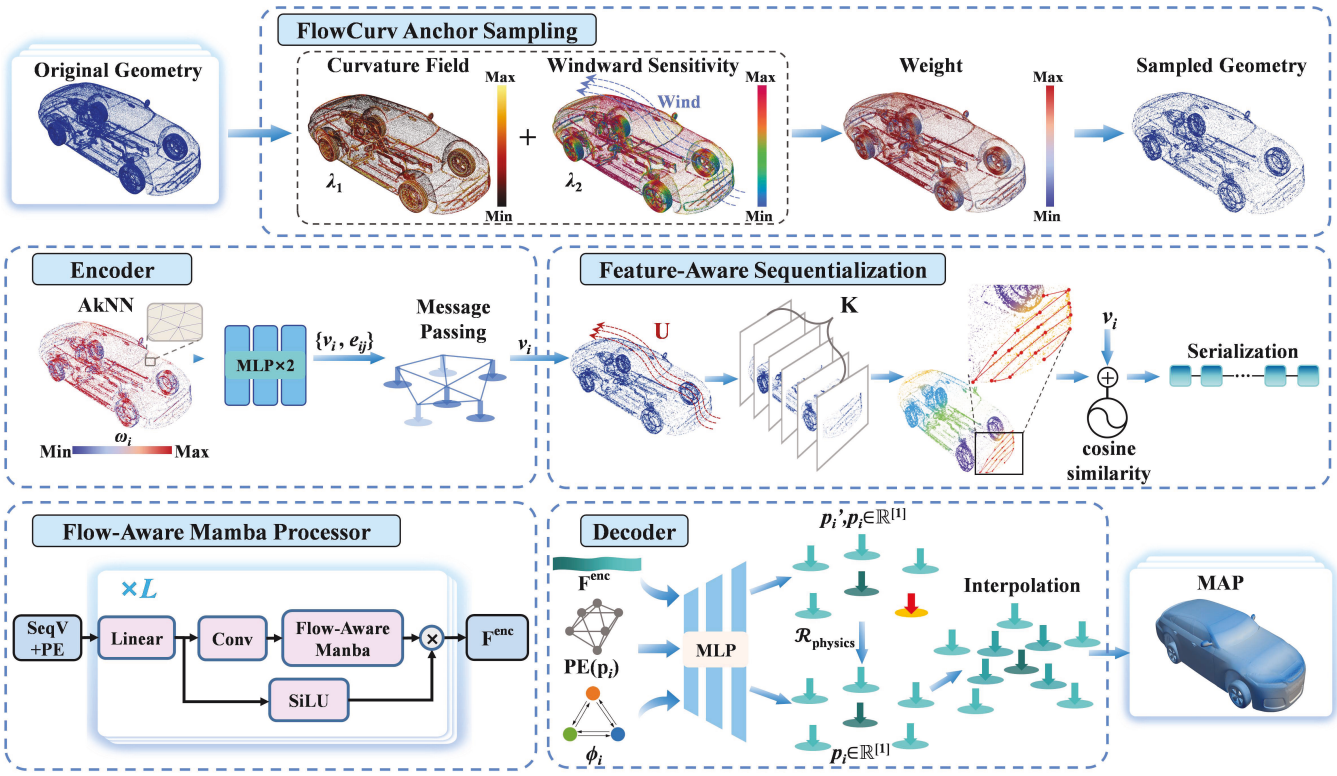

Figure 1: Overview of the FCMO architecture, integrating flow-curv Mamba operations for efficient modeling of extensive 3D vehicle aerodynamic simulations. From an input 3D vehicle mesh, the system delivers precise predictions for key surface metrics including pressure distributions, wall shear stresses, and drag coefficients.

et al. 2025) integrates localized feature learning via message propagation with global dependency modeling through projection-based attention, utilizing frequency-augmented graph architectures with nearest neighbor connectivity for accurate local representation.

Methodology

We propose FCMO, a flow-curv Mamba operator specifically designed for solving large-scale CFD problems on complex geometries. This section details its architecture, beginning with problem formulation and notation, followed by core components description, and concluding with complexity analysis.

Problem Setting and Notations

Neural operators aim to learn mappings between infinite-dimensional function spaces. Let $\mathcal{A}$ represent the input function space and $\mathcal{S}$ the solution space, where the neural operator seeks to learn an operator $\mathcal{F} : \mathcal{A} \to \mathcal{S}$. We focus on constructing a parameterized neural network $\mathcal{F}_\theta$ to approximate this operator, which maps arbitrary 3D vehicle geometries with given boundary conditions to their surface physical fields (e.g., pressure). For a given k-th vehicle instance, the input a^k consists of two components: (1) the vehicle geometry discretized by a 3D mesh $\mathcal{G}^k = \{\mathcal{P}^k, \mathcal{L}^k\}$, where $\mathcal{P}^k = \{(x_i^k, y_i^k, z_i^k)\}_{i=1}^M$ represents the 3D coordinates of M vertices and $\mathcal{L}^k$ denotes the mesh edges; (2) boundary conditions $\mathcal{B}^k \in \mathbb{R}^m$ containing global design parameters such as vehicle dimensions, Reynolds number, and inlet velocity. The expected output is the discretized surface pressure $s^k = \{q_j^k\}_{j=1}^M$, where $q_j^k \in \mathbb{R}$ is the pressure value at the j-th node. Following conventional neural operator approaches, our objective is to minimize the L_2 relative error loss between predictions $\hat{s}^k$ and ground truth s^k on the training dataset:

$$\min_\theta \frac{1}{D} \sum_{k=1}^{D} \frac{\|s^k - \hat{s}^k\|_2}{\|s^k\|_2} \tag{1}$$

where D is the training dataset size and θ represents the network's trainable parameters.

Model Architecture

As shown in Figure 1, the core insight of FCMO stems from fluid physics: flow information propagation exhibits significant directionality and causality. The framework implements a unified processing pipeline comprising three key components: spatial feature encoder, flow-aware Mamba processor, and physics-constrained decoder. Mamba's selective mechanism naturally aligns with these fluid dynamics principles, with theoretical justification provided in Appendix A.

Encoder The encoder samples critical nodes from large-scale raw meshes and extracts feature representations rich in

local geometric and physical information.

FlowCurv Anchor Sampling. To reduce computational complexity while preserving critical physical regions, we propose a geometry-physics adaptive sampling strategy. We compute a physical importance weight w_i for each vertex i, which fuses normalized local curvature κ_i and windward sensitivity $c_i = 1 - |\mathbf{n}_i \cdot \mathbf{U}|$, where $\mathbf{n}_i$ is the surface unit normal vector at mesh vertex $\mathbf{p}_i$ and $\mathbf{U}$ is the free-stream velocity direction:

$$w_i = \lambda_1 \kappa_i + \lambda_2 c_i \tag{2}$$

Subsequently, we perform two-stage sampling based on these weights: first, probabilistic sampling retains high-weight anchor points, then improved farthest point sampling completes the remaining nodes, totaling N sampled points ($N \ll M$). The distance metric during sampling is modified to:

$$d(i,j) = \frac{\|\mathbf{p}_i - \mathbf{p}_j\|_2}{\sqrt{w_i + w_j}} \tag{3}$$

This metric scales geometric distance through physical importance weights, enabling dense sampling in critical regions.

Adaptive Feature Extraction. After geometry-physics adaptive sampling, we obtain a set of non-uniformly distributed key nodes. This stage includes two core steps: adaptive neighborhood construction and dual-scale position encoding.

First, we employ an adaptive k-nearest neighbor method to construct graph neighborhood relationships. Since fixed neighbor counts cannot adapt to dramatic sampling density variations, we dynamically adjust the neighbor count k_i for each node based on its physical importance weight w_i:

$$k_i = k_{\text{base}} + \beta \left\lfloor \frac{w_i}{\bar{w}} \right\rfloor \tag{4}$$

where k_{base} is the base neighbor count, $\bar{w}$ is the average weight, and β is a scaling factor controlling the influence strength. This design enables nodes in physically critical regions to have more neighbors for capturing complex details, while nodes in flat regions connect to fewer neighbors.

Second, to enable the model to perceive multi-scale physical phenomena, we design a novel dual-scale physics-aware position encoding (PE):

$$\text{PE}(\mathbf{p}_i) = \text{SPE}(\mathbf{p}_i) + w_i \cdot \text{SPE}\left(\frac{\mathbf{p}_i}{L_{\text{ref}}}\right) \tag{5}$$

This encoding consists of two components: the first term $\text{SPE}(\mathbf{p}_i)$ provides global, absolute coordinate information; the second term scales coordinates by a reference length L_{ref} to generate a fine-scale encoding highly sensitive to minute positional changes, weighted by the physical importance weight w_i.

Based on this structure, the initial node feature vector $\mathbf{v}_i^{(0)}$ is encoded through two independent multi-layer perceptrons, fusing coordinates, boundary conditions, position encoding, and physical importance information. Edge features integrate edge length, orientation, and node weights:

$$\begin{aligned} \mathbf{v}_i^{(0)} &= \Phi^V(\mathbf{p}_i, \mathcal{B}, \text{PE}(\mathbf{p}_i), w_i) \\ \mathbf{e}_{ij} &= \Phi^E(\|\mathbf{p}_i - \mathbf{p}_j\|_2, \mathbf{p}_i - \mathbf{p}_j, w_i + w_j) \end{aligned} \tag{6}$$

Finally, we aggregate local information and update node features through an L-layer graph neural network (GNN). To prevent spatial information decay due to multiple message passing in deep networks, we reintroduce position encoding (PE) in each layer's update formula:

$$\mathbf{v}_i^{(l+1)} = \mathbf{v}_i^{(l)} + \Phi_P^{(l)}\left(\mathbf{v}_i^{(l)}, \sum_{j \in \mathcal{N}_i} \mathbf{e}_{ij}^{(l)}, \mathbf{v}_j^{(l)}, \text{PE}(\mathbf{p}_i)\right) \tag{7}$$

This design ensures nodes maintain global position awareness throughout the learning process. The encoder outputs a spatially enhanced node feature matrix $\mathbf{V} \in \mathbb{R}^{N \times d}$ after L iterations.

Processor The processor serves as the core of FCMO, converting encoded spatial features into one-dimensional sequences and capturing long-range dependencies along flow directions through improved Mamba models.

Feature-Aware Sequentialization. This module converts graph data into 1D sequences while preserving physical causality in fluid flows. Traditional scanning methods often disrupt spatial relationships in irregular meshes, leading to suboptimal long-range dependency modeling. To overcome this, we design a stratified, feature-guided approach that first establishes a global flow-aligned order and then refines it locally for enhanced coherence.

We begin by using the free-stream direction $\mathbf{U}$ as the primary sorting axis, stratifying nodes based on their projections onto the $\mathbf{U}$ axis to create an initial global upstream-downstream ordering $\mathcal{L}^{\text{init}}$. This ensures that the sequence respects the natural progression of aerodynamic phenomena.

Next, to cluster physically similar nodes and refine the sequence for better Mamba processing, we apply local adjustments using node feature cosine similarity:

$$\mathcal{L}_k^{\text{final}} = \mathcal{L}_k^{\text{init}} \cup \{(\mathbf{p}_j, \mathbf{v}_j) : \text{sim}(\mathbf{v}_j, \mathbf{v}_k) > \tau, \mathbf{p}_j \in \mathcal{L}^{\text{init}}\} \tag{8}$$

where τ is the similarity threshold, $\mathbf{p}_j$ is the position of node j, $\mathbf{v}_j$ is its feature vector, and k indexes the layers. This step groups nodes with akin physical properties, fostering a more meaningful sequence.

The number of layers K is adaptively set to balance detail and efficiency, based on mesh resolution and geometric complexity:

$$K = \max\left(5, \min\left(20, \left\lfloor \frac{L_{\text{span}}}{L_{\text{ref}}} \cdot \sqrt{N} \right\rfloor\right)\right) \tag{9}$$

where L_{span} is the vehicle's projection length along the flow direction, L_{ref} is a reference length (e.g., vehicle width), and N is the number of nodes.

Within each layer, we implement a feature-guided snake scanning strategy to maintain local continuity. The sorting weight integrates geometric positions with features:

$$w_{\text{sort}}(\mathbf{p}_i) = \alpha \cdot \mathbf{d}_{\text{perp}} \cdot \mathbf{p}_i + (1 - \alpha) \cdot \text{proj}(\mathbf{v}_i) \tag{10}$$

where $\mathbf{d}_{\text{perp}}$ is a vector perpendicular to $\mathbf{U}$, α is the geometric weight (typically 0.7), $\mathbf{p}_i$ and $\mathbf{v}_i$ are the position and feature of node i, and $\text{proj}(\cdot)$ is a learnable projection function.

By concatenating these optimized layers, we produce a feature sequence SeqV that robustly preserves spatial locality and physical causality, enabling Mamba to capture intricate long-range flow dependencies effectively.

Flow-Aware Mamba Processor. This processor employs improved state space models (SSM) to process sequentialized features. We extend the standard Mamba architecture to make its selective mechanism physics-aware. Specifically, state space parameters $\mathbf{B}_t$ and $\mathbf{C}_t$ depend not only on input features $\mathbf{x}_t$ and spatial step size Δ_t, but also on critical physical features $\boldsymbol{\phi}_t$ at the current position:

$$\begin{aligned} \mathbf{B}_t &= s_B(\mathbf{x}_t, \Delta_t, \boldsymbol{\phi}_t), \quad \mathbf{C}_t = s_C(\mathbf{x}_t, \Delta_t, \boldsymbol{\phi}_t) \\ \mathbf{F}^{\text{enc}} &= \text{FCMO}_L(\text{FCMO}_1(\text{SeqV} + \text{PE})) \end{aligned} \tag{11}$$

This design enables the model to adaptively adjust its receptive field, performing fine-grained modeling in regions where physical quantities change dramatically, such as boundary layers. Through stacking L FCMO layers, the model effectively learns complex long-range dependencies.

Decoder The decoder maps processor output sequence features $\mathbf{F}^{\text{enc}}$ back to physical space and applies physical constraints to ensure solution reasonability. The predicted pressure value q_i for node $\mathbf{p}_i$ is generated by an MLP decoder $\mathcal{D}$, whose input fuses sequence features, position encoding, and key physical features:

$$\begin{aligned} q_i^{\text{final}} &= \mathcal{D}(\mathbf{F}^{\text{enc}}, \text{PE}(\mathbf{p}_i), \boldsymbol{\phi}_i) \\ q_i &= q_i^{\text{final}} + \lambda_p \cdot \mathcal{R}_{\text{physics}}(q_i^{\text{final}}, \mathcal{N}_i) \end{aligned} \tag{12}$$

where $\mathcal{R}_{\text{physics}}$ utilizes local neighborhood information $\mathcal{N}_i$ of node $\mathbf{p}_i$ to ensure local pressure gradient smoothness and enforce known boundary conditions. The hyperparameter λ_p controls the strength of physical constraints.

For unsampled points in the computational domain, we employ a hybrid weighted interpolation method that considers both Euclidean distance to surrounding predicted points and GNN feature similarity, smoothly extending these physically corrected predictions to the entire flow field mesh.

Training Objective and Complexity Analysis We predict surface pressure only and enforce incompressibility and boundary conditions through physics-based regularization.

$$\mathcal{L}_{\text{total}} = L_2(\hat{p},\, p) + \lambda_{\text{div}} \left\|\nabla \cdot \mathbf{u}\right\|_2^2 + \lambda_{\text{bc}}\, \mathcal{L}_{\text{BC}}. \tag{13}$$

The first term measures the data fidelity between the decoder output $\hat{p}$ and the ground-truth pressure p. The second and third terms impose incompressibility and boundary conditions, weighted by λ_{div} and λ_{bc}, respectively.

$$\mathbf{u} \;=\; \mathcal{R}\big(\hat{p};\, \Omega,\, U_\infty\big). \tag{14}$$

The auxiliary velocity $\mathbf{u}$ used in the physics losses is reconstructed from the predicted pressure $\hat{p}$ via a differentiable pressure–projection operator $\mathcal{R}$ on domain Ω with freestream U_∞; no separate velocity head is required.

$$\mathcal{L}_{\text{BC}} = \frac{1}{|\Gamma_w|} \sum_{i \in \Gamma_w} \left\|\mathbf{u}_i\right\|_2^2 + \tau\, \frac{1}{|\Gamma_{\text{far}}|} \sum_{j \in \Gamma_{\text{far}}} \left\|\mathbf{u}_j - U_\infty\right\|_2^2. \tag{15}$$

here Γ_w denotes no-slip surfaces and Γ_{far} denotes the outer (freestream) boundary used by $\mathcal{R}$. The scalar γ balances wall and far-field penalties.

In analyzing the overall computational complexity of FCMO, we focus on its main components: adaptive sampling, graph neural network processing, and Mamba sequence modeling. Given the original mesh vertex count M, sampled node count N, graph edge count E, feature dimension d, and processor layer count L, the adaptive sampling has complexity $\mathcal{O}(M \log N)$, the GNN encoder requires $\mathcal{O}(Ed)$, and the Mamba processor scales as $\mathcal{O}(NdL)$. Benefiting from Mamba's linear complexity, FCMO avoids the $\mathcal{O}(N^2)$ bottleneck of Transformer architectures. The overall complexity is $\mathcal{O}_{\text{total}} = \mathcal{O}(Ed + NdL)$, where $N \ll M$ due to adaptive sampling, providing significant efficiency advantages for large-scale 3D geometries.

Experiments and Results

To validate the effectiveness of FCMO for complex and large-scale aerodynamics learning tasks in automotive applications, we conduct comprehensive experiments on two challenging industry-standard benchmarks with various geometries.

Experimental Setup and Evaluation Metrics

Datasets. We evaluate FCMO using two benchmark datasets for vehicle aerodynamics, summarized in Table 1. Ahmed-Body (Li, Shu, and Barati Farimani 2023) provides simplified vehicle geometries with varying slant angles, while DrivAerNet (Elrefaie, Dai, and Ahmed 2024b) offers realistic car models with 26-parameter geometric variations and underbody details. We randomly sample 550 geometries from DrivAerNet due to computational constraints. Both datasets contain CFD simulations at Reynolds numbers up to 5 million, with ground truth measurements for surface pressure, wall shear stress, and aerodynamic coefficients.

Dataset	**Ahmed-Body**	**DrivAerNet**
Total geometries	551	4,000
Used samples (train/test)	500/51	500/50*
Surface mesh faces	0.12M	0.5M
Design parameters	Slant angle	26 parameters
Vehicle type	Simplified	Realistic†
Reynolds number	Up to $5{\times}10^6$	Up to $5{\times}10^6$

* Randomly selected subset. † With wheels and underbody.

Table 1: Benchmark datasets for vehicle aerodynamics.

Baselines. We compare FCMO with nine competitive baselines spanning different architectures. Graph-based methods include GAT (Veličković et al. 2018), EGNNs (Satorras, Hoogeboom, and Welling 2021), and MeshGraphNet (MGN) (Pfaff et al. 2020). Transformer-based neural operators comprise GNOT (Hao et al. 2023), IPOT (Lee and Oh 2024), Transolver (Wu et al. 2024), and GINO (Li et al. 2023). Recent advances include GTAT (Authors 2025) and AeroGTO (Liu et al. 2025). All methods use identical preprocessing and hyperparameter optimization.

Methods	Ahmed-Body				DrivAerNet			
	Pressure ↓	**Wall Shear** ↓	C_d ↓	R_s ↑	**Pressure** ↓	**Wall Shear** ↓	C_d ↓	R_s ↑
GAT (2018)	0.1388	0.1876	0.1041	0.9256	0.2254	0.2834	0.1850	0.9123
EGNNs (2021)	0.1356	0.1823	0.1028	0.9342	0.2198	0.2756	0.1823	0.9198
MGN (2020)	0.1367	0.1845	0.1035	0.9298	0.2216	0.2789	0.1834	0.9156
GNOT (2023)	0.1293	0.1734	0.1002	0.9378	0.2471	0.2987	0.1892	0.9067
IPOT (2024)	0.1003	0.1456	0.0755	0.9687	0.1934	0.2398	0.1567	0.9534
GINO (2023)	0.0831	0.1287	0.0532	0.9823	0.1618	0.2087	0.1245	0.9756
Transolver (2024)	0.0927	0.1354	0.0679	0.9751	0.1719	0.2234	0.1389	0.9672
GTAT (2025)	0.1245	0.1698	0.0967	0.9421	0.2087	0.2567	0.1756	0.9287
AeroGTO (2025)	0.0757	0.1198	0.0475	0.9876	0.1524	0.1934	0.1089	0.9847
FCMO (ours)	**0.0718**	**0.1087**	**0.0421**	**0.9935**	**0.1445**	**0.1798**	**0.0987**	**0.9918**

Table 2: Performance comparison on two large-scale benchmarks.

Metrics	GAT	EGNNs	MGN	GNOT	IPOT	Transolver	GINO	GTAT	AeroGTO	FCMO
N_P/M ↓	2.35	2.34	2.33	3.66	6.73	2.84	230.69	2.57	2.44	**2.32**
FLOPs/GFLOPs ↓	502.0	487.2	506.4	381.6	285.2	321.1	30839.3	452.9	175.3	**158.7**
GM (GB) ↓	18.5	17.8	19.2	22.4	28.9	20.1	76.3	16.7	14.2	**12.8**

Table 3: Computational efficiency comparison. Lower values indicate better efficiency.

Evaluation Metrics. We assess model performance across multiple physical quantities and computational efficiency. For surface pressure prediction, we use mean L_2 relative error. Wall shear stress prediction is evaluated using L_2 relative error. For aerodynamic coefficients, we measure mean relative error (MRE) for drag coefficient (C_d) estimation, where C_d is computed as $C_d = \frac{2}{v^2 A}\int_S p(x)(n(x)\cdot i)dx$, with i representing the streamwise direction. Additionally, we evaluate Spearman's rank correlation coefficient (R_s) for drag coefficient predictions to assess the model's capability to rank different vehicle designs for aerodynamic optimization. Computational efficiency metrics include number of trainable parameters (N_P), floating-point operations (FLOPs), and GPU memory usage (GM).

Implementation Details. All experiments are conducted on NVIDIA A100 GPUs with 40GB memory. We use Adam optimizer with learning rate 1×10^{-4}, weight decay 1×10^{-5}, and cosine annealing scheduler. The model is trained for 200 epochs with batch size 8. For FCMO configuration, FlowCurv sampling ratio is set to 0.3 for Ahmed-Body and 0.2 for DrivAerNet, physical importance weights $\lambda_1 : \lambda_2 = 2 : 1$, adaptive k-NN with $k_{base} = 16$ and $\beta = 0.5$, reference length $L_{ref} = L_{vehicle}/10$, 6 Mamba layers with feature dimension 128, and physical constraint weight $\lambda_{phys} = 0.1$. Results are averaged over three independent runs.

Comparison with State-of-the-Art Methods

Multi-Physics Field Prediction Performance Table 2 presents the main results of multi-physics field predictions on test datasets. FCMO achieves consistent state-of-the-art performance across all evaluation metrics on both benchmarks. Compared to the previous best method AeroGTO, FCMO delivers improvements of 5.2% in surface pressure prediction on both Ahmed-Body (0.0718 vs 0.0757) and DrivAerNet (0.1445 vs 0.1524). For wall shear stress prediction, FCMO achieves gains of 9.3% on Ahmed-Body and 7.0% on DrivAerNet, demonstrating the effectiveness of our FlowCurv sampling and dual-scale positional encoding. FCMO also excels in aerodynamic coefficient estimation, improving C_d prediction by 11.4% on Ahmed-Body and 9.4% on DrivAerNet. Furthermore, FCMO achieves excellent design ranking accuracy with R_s of 0.9935 and 0.9918 respectively, validating the flow-aware Mamba processor's capability to capture complex aerodynamic correlations. Figure 2 presents qualitative comparisons showing FCMO's significantly reduced prediction errors, particularly in critical aerodynamic regions such as vehicle front surfaces and wake regions.

Computational Efficiency Analysis Table 3 demonstrates FCMO's superior computational efficiency compared to existing methods. FCMO maintains competitive parameter efficiency with 2.32M parameters while achieving 9.4% reduction in FLOPs compared to AeroGTO (158.7 vs 175.3 GFLOPs) and remarkable 99.5% reduction compared to GINO. Memory efficiency is notable, with FCMO consuming only 12.8 GB peak GPU memory, representing 9.9% improvement over AeroGTO and 83.2% reduction compared to GINO. This efficiency stems from our adaptive sampling strategy and Mamba's linear complexity that avoids the quadratic bottleneck of attention mechanisms. The results affirm FCMO as an efficient solution for large-scale automotive aerodynamics, with Fig. 3 showing optimal tradeoffs across both datasets.

Ablation Studies

We conduct comprehensive ablation studies to validate the effectiveness of each proposed component in FCMO. The experiments systematically evaluate six key modules: FlowCurv Sampling (FCS), Adaptive k-NN construction (AkNN), Dual-scale Physics-aware Position Encod-

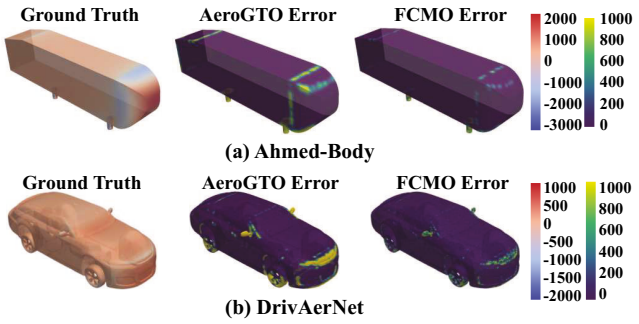

Figure 2: Qualitative comparison of surface pressure field prediction errors.

Dataset Method	Ahmed-Body L_2 Error↓	DrivAerNet L_2 Error↓
FCMO (Full)	**0.0718**	**0.1445**
FCMO w/o FCS	0.0834	0.1623
FCMO w/o AkNN	0.0789	0.1534
FCMO w/o DPE	0.0856	0.1598
FCMO w/o FAS	0.0912	0.1687
FCMO w/o FAM	0.1045	0.1834
FCMO w/o PC	0.0751	0.1489

Table 4: Ablation study on pressure field prediction across two datasets. Complete ablation results are provided in Appendix B.

ing (DPE), Feature-Aware Sequentialization (FAS), Flow-Aware Mamba processor (FAM), and Physics Constraints (PC). Each component is removed individually to assess its contribution on both benchmarks.

As presented in Table 4, all proposed components work synergistically to achieve optimal performance. FAM exhibits the most substantial contribution, with its removal resulting in performance degradation of 45.5% and 26.9% on Ahmed-Body and DrivAerNet respectively, confirming the critical importance of physics-aware selective state space modeling. FAS demonstrates the second-largest impact with 27.0% and 16.7% performance drops, validating our causality-preserving sequence construction. DPE contributes 19.2% and 10.6% improvements, demonstrating that multi-scale spatial awareness is essential for accurate aerodynamic modeling. While PC shows the smallest individual impact with 4.6% and 3.0% improvements, it provides valuable regularization for physical consistency.

Out-of-Distribution Generalization

To evaluate FCMO's generalizability beyond training distributions, we conduct out-of-distribution (OOD) experiments using test samples from DrivAerNet++ (Elrefaie, Dai, and Ahmed 2024a) with parameter ranges completely outside the training data. We evaluate two OOD scenarios: Reynolds number tests with $Re \in [1.8 \times 10^5, 2.5 \times 10^5] \cup [1.5 \times 10^6, 2.2 \times 10^6]$ versus training range $Re \in [2.5 \times 10^5, 1.5 \times 10^6]$, and vehicle scale tests with $L_{char} \in [3.2m, 3.8m] \cup [4.8m, 5.6m]$ versus training range $L_{char} \in [3.8m, 4.8m]$.

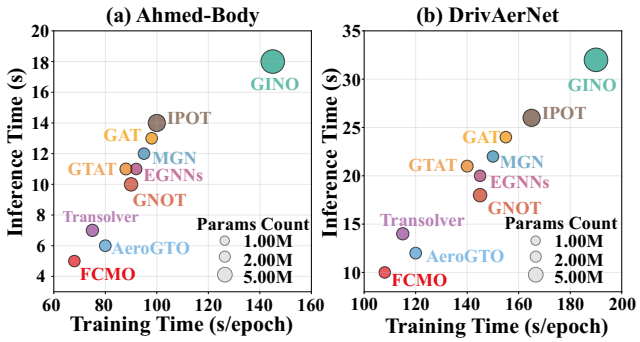

Figure 3: Computational efficiency comparison on Ahmed-Body and DrivAerNet datasets.

As shown in Table 5, FCMO consistently achieves superior OOD performance across both scenarios, maintaining drag coefficient mean relative errors of 0.0634 and 0.1289 with Spearman correlations of 0.9758 and 0.9684 respectively. Despite the inherent performance degradation under challenging OOD conditions, the flow-aware Mamba processor's physics-informed selective mechanism effectively captures generalizable aerodynamic principles, while FlowCurv sampling preserves critical flow features across varying operating conditions, demonstrating FCMO's practical reliability for real-world automotive design applications with superior generalization capabilities.

Methods	OOD Reynolds		OOD Vehicle Scale	
	MRE ↓	R_s ↑	MRE ↓	R_s ↑
GAT	0.2234	0.8567	0.2789	0.8234
EGNNs	0.2167	0.8634	0.2712	0.8298
MGN	0.2198	0.8598	0.2743	0.8267
GNOT	0.2089	0.8723	0.2634	0.8356
IPOT	0.1756	0.9012	0.2134	0.8767
Transolver	0.1534	0.9178	0.1923	0.8934
GINO	0.1398	0.9267	0.1789	0.9045
AeroGTO	0.1267	0.9456	0.1567	0.9234
FCMO (ours)	**0.0634**	**0.9758**	**0.1289**	**0.9684**

Table 5: Out-of-distribution generalization results on DrivAerNet++ datasets.

Conclusion

Large-scale automotive aerodynamics faces critical challenges in complex physics modeling and computational scalability. We propose FCMO, a flow-aware Mamba operator that integrates selective state space mechanisms for efficient vehicle aerodynamic prediction. Through processing 3D vehicle meshes, FCMO demonstrates robust capabilities in surface pressure prediction and drag coefficient estimation. Comprehensive experiments validate FCMO's superior performance across two industry-standard benchmarks against nine state-of-the-art methods.

Acknowledgments

This work was supported by the Natural Science Foundation of Jiangsu Province under Grant (No.BK20240663), and the National Natural Science Foundation of China under Grant (No.62501287, No.62272242).

References

Ahmed, S. R.; Ramm, G.; and Faltin, G. 1984. Some salient features of the time-averaged ground vehicle wake. *SAE Technical Paper*, (840300).

Authors, M. 2025. GTAT: empowering graph neural networks with cross attention. *Scientific Reports*, 15(2025). Graph Topology Attention Networks.

Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; and Vandergheynst, P. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42.

Cao, Y.; Chai, M.; Li, M.; and Jiang, C. 2023. Efficient learning of mesh-based physical simulation with bi-stride multi-scale graph neural network. In *Proceedings of the 40th International Conference on Machine Learning*, 3541–3558.

Dao, T.; and Gu, A. 2024. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*.

Dhaubhadel, M. N. 1996. CFD applications in the automotive industry. *Journal of Fluids Engineering*, 118(4): 647–653.

Elrefaie, M.; Dai, A.; and Ahmed, F. 2024a. DrivAerNet++: A Large-Scale Multimodal Car Dataset with Computational Fluid Dynamics Simulations and Deep Learning Benchmarks. In *Advances in Neural Information Processing Systems*, volume 37.

Elrefaie, M.; Dai, A.; and Ahmed, F. 2024b. Drivaernet: A parametric car dataset for data-driven aerodynamic design and graph-based drag prediction. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 88360, V03AT03A019. American Society of Mechanical Engineers.

Gladstone, R. J.; Rahmani, H.; Suryakumar, V.; et al. 2024. Mesh-based GNN surrogates for time-independent PDEs. *Scientific Reports*, 14: 3394.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Hao, Z.; Wang, Z.; Su, H.; Ying, C.; Dong, Y.; Liu, S.; Cheng, Z.; Song, J.; and Zhu, J. 2023. GNOT: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, 12556–12569. PMLR.

Kochkov, D.; Smith, J. A.; Alieva, A.; Wang, Q.; Brenner, M. P.; and Hoyer, S. 2021. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21): e2101784118.

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

Lee, M.; and Moser, R. D. 2015. Direct numerical simulation of turbulent channel flow up to $\mathrm{Re}_\tau = 5200$. *Journal of Fluid Mechanics*, 774: 395–415.

Lee, S.; and Oh, T. 2024. Inducing point operator transformer: A flexible and scalable architecture for solving PDEs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 153–161.

Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; and Anandkumar, A. 2020. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*.

Li, Z.; Kovachki, N.; Choy, C.; Li, B.; Kossaifi, J.; Otta, S. P.; Nabian, M. A.; Stadler, M.; Hundt, C.; Azizzadenesheli, K.; and Anandkumar, A. 2023. Geometry-informed neural operator for large-scale 3D PDEs. In *Advances in Neural Information Processing Systems*, volume 36, 35836–35854.

Li, Z.; Shu, D.; and Barati Farimani, A. 2023. Scalable transformer for pde surrogate modeling. *Advances in Neural Information Processing Systems*, 36: 28010–28039.

Liu, J.; Wang, N.; Wang, L.; et al. 2024a. Latent neural operator for solving forward and inverse PDE problems. In *Advances in Neural Information Processing Systems*.

Liu, P.; Wang, P.; Ren, X.; Yuan, H.; Hao, Z.; Xu, C.; Cai, S.; and Ni, D. 2025. AeroGTO: An efficient graph-transformer operator for learning large-scale aerodynamics of 3D vehicle geometries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39.

Liu, P.; et al. 2024b. State-space models are accurate and efficient neural operators for dynamical systems. *arXiv preprint arXiv:2409.03231*.

O'Leary-Roseberry, T.; Chen, P.; and Ghattas, O. 2024. Derivative-informed neural operator: An efficient framework for high-dimensional parametric derivative learning. *Journal of Computational Physics*, 496: 112555.

Pfaff, T.; Fortunato, M.; Sanchez-Gonzalez, A.; and Battaglia, P. W. 2020. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 5099–5108.

Rusch, T. K.; Bronstein, M. M.; and Mishra, S. 2023. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*.

Satorras, V. G.; Hoogeboom, E.; and Welling, M. 2021. E(n) equivariant graph neural networks. In *International Conference on Machine Learning*, 9323–9332. PMLR.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1): 61–80.

Taghizadeh, L.; Karpatne, A.; and Luo, Z. 2025. Multifidelity graph neural networks for efficient and accurate mesh-based partial differential equations surrogate modeling. *Computer-Aided Civil and Infrastructure Engineering*, 40: 220–238.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Wang, S.; Wang, H.; and Perdikaris, P. 2021. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances*, 7(40): eabi8605.

Wu, H.; Luo, H.; Wang, H.; Wang, J.; and Long, M. 2024. Transolver: A fast transformer solver for PDEs on general geometries. *arXiv preprint arXiv:2402.02366*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Zhang, E.; Kahana, A.; Kopaničáková, A.; et al. 2024. Blending neural operators and relaxation methods in PDE numerical solvers. *Nature Machine Intelligence*, 6: 1303–1313.