A Review of AB-UPT: Anchored-Branched Universal Physics Transformers for Automotive Aerodynamics CFD Surrogate Modeling

Justin Feldman, FE
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
feldman.jus@northeastern.edu

Abstract

Anchored-Branched Universal Physics Transformer, or AB-UPT [1], is a physics informed neural surrogate model designed to address the problem scale complexity of computational fluid dynamics, or CFD, in the domain of automotive aerodynamics. Neural surrogate research in this domain has largely been unsuccessful to create a viable model that provides the requisite accuracy to implement in practice without trading off for computational cost. AB-UPT sets the standard for accuracy among other existing methods while being comparable or beating them in training and inference efficiency.

1 Introduction

CFD is the computer simulation of fluid flow behaviors as grounded by the core fluid dynamics physics formulas, the Navier-Stokes equations. CFD is used to determine the lift force on airplane wings to determine material constraints and plane maneuverability. CFD is used in biomedical industries to determine intravenous flow behaviors or to optimize a device's microfluidics design. CFD is used to determine airflow laminarity in biopharmaceutical clean rooms. CFD is a core tool that so many critical engineering industries rely on.

As significant as CFD is, it is just as expensive and complex to properly implement. As referenced in the original AB-UPT paper, their dataset of interest, DrivAerML, was produced using the industry gold standard CFD simulation algorithm necessary for critical engineering applications. DrivAerML took roughly $1M dollars and 826 days of effective compute time to generate just 484 simulations.

This combination of practical necessity and high computational cost has attracted an abundance of neural surrogate research to create a lightweight, yet representative, model of CFD. Existing methods have struggled to balance the tradeoff of accuracy and efficiency in the highly complex solution space of fluid dynamic flows and the scale of the input features given by the 3D object's geometry and the volumetric meshes used in CFD simulations. As a mechanical engineer who has performed numerous CFD simulations, I am intimately familiar with this problem.

AB-UPT introduces an elegant, novel, architecture to approach the scaling bottleneck while generating significantly more accurate solutions than other existing models. The core 3 contributions are as follows:

- **A multi-branch architecture**, decoupling the encoding block from the decoding block.

Preprint.

- **Anchor-attention**, vastly reducing the number of tokens that undergo the $\mathcal{O}(n^2)$ self-attention operation.
- **Hard physics constraints**, directly applying physics formulas to guarantee physics consistent solutions.

The multi-branched architecture enables, generalizable, CAD-only inference and, anchor-attention makes hard constraints tractable via the conditional neural field output.

This review summarizes AB-UPT's core methods and findings while adding context on physics priors, the model's understated benefits, and comparable architectures in CFD surrogate research.

2 Approach

2.1 CFD Background & Problem Formulation

Alkin et al. assume their audience has domain expertise in the field of fluid dynamics. I'll expand on the paper's physics terminology to further justify the value of a surrogate model and validate why AB-UPT's design choices were made to those without it. While the CFD process is a full subfield of study to fluid dynamics, I will review the following, most relevant components of CFD with respect to AB-UPT: the Navier-Stokes equations, RANS, LES, and HRLES, and the finite-volume meshing (FVM) process.

2.1.1 The Navier-Stokes Equations

The Navier-Stokes equations formulate the relationship between velocity and pressure in viscous fluids (liquids and gasses). The equations were derived independently by G.G. Stokes and M. Navier in the early 1800's. These equations, while complex, are effectively Newton's Laws of physics with respect to fluids.

The conservation of mass and momentum equations are given by:

$$\nabla \cdot u = 0 \tag{1}$$

$$\rho \left(\frac{\partial u}{\partial t} + u \cdot \nabla u \right) = -\nabla p + \mu \nabla^2 u + F \tag{2}$$

Equation (1) constrains the divergence of a fluid's velocity in a reference volume to 0. ∇, is the gradient of u, the 3D velocity vector of the fluid. The gradient describes the rate of change of the fluid's velocity, or acceleration, in all directions. Practically, this equation states that the fluid entering a reference volume is equivalent to the fluid exiting a reference volume without fluid sources or sinks.

Equation (2) states that the momentum, mass by velocity, of the system is conserved absent external forces. Analogize conservation of momentum to the collision of two bowling balls (given a fully elastic collision without any friction forces). The momentum of the bowling balls before the collision is equal to the momentum of the bowling balls after the collision. A fluid is comprised of molecules, or per our analogy, bowling balls. The left-hand side of the equation represents mass by acceleration, or force imparted by the entire fluid body. The right-hand side of the equation represents the internal forces of the fluid (all the colliding bowling balls) summed with the external forces acting upon the fluid, like gravity.

The first internal force term, ∇p, is the rate of change of pressure in the fluid. Fluid moves from areas of high pressure to low pressure, and this term captures that behavior. The second internal force term, $\mu \nabla^2 u$, represents the fluid's viscosity, or the fluid's internal friction. Honey has a high viscosity while air has a low viscosity. As previously stated, the last term, F, represents external forces applied to the fluid.

2.1.2 RANS vs LES vs HRLES

Fluids are turbulent when their convective forces overwhelm their viscous forces. Turbulence is a highly energetic and chaotic phenomenon that you have likely experienced on a bumpy airplane ride. While turbulence may seem like a random phenomenon, it is calculable by the Navier-Stokes equations. However, the scale at which turbulent behavior dissipates into thermal energy loss is on

the Kolmogorov scale, typically measured in mm's, which is far too fine of a scale to practically simulate fluid flow for industrial applications. Averaged formulations such as RANS, LES, and their hybrid, HRLES, approach this issue and are used in practice to conduct simulations, specifically, the simulations that AB-UPT trains on [7].

RANS. The Reynolds decomposition, mean flow, and turbulent fluctuations are defined as follows:

$$u(\mathbf{x}, t) = \bar{U}(\mathbf{x}) + u'(\mathbf{x}, t) \tag{3}$$

$$\bar{U}(\mathbf{x}) = \lim_{T \to \infty} \int_0^T u(\mathbf{x}, t)\, dt \tag{4}$$

$$u'(\mathbf{x}, t) = u(\mathbf{x}, t) - \bar{U}(\mathbf{x}) \tag{5}$$

In RANS, or Reynolds Averaged Navier-Stokes, the velocity vector, $u(x, t)$, is decomposed into an average component, $\bar{U}(\mathbf{x})$, and fluctuating component, $u'(\mathbf{x}, t)$. The average component is formulated by averaging out the u-velocity vector as time, T, goes to infinity across each point in the reference volume. The fluctuating components are the residuals between the true vector field and the averaged field. This Reynold's decomposition effectively creates a statistical distribution of the fluid flow's velocity at any given location at any given time in the flow's reference volume.

LES. In LES, or Large Eddy Simulations, instead of solving the Navier-Stokes equations in a high-resolution domain, LES solves the Navier-Stokes equations over a coarsened grid of the domain. While this coarsened grid still introduces sub-grid-scale challenges that require models themselves, it produces more accurate representations than RANS in exchange for slightly more computational cost.

HRLES. HRLES, or Hybrid RANS-LES, is the gold standard used in industry. It uses RANS near boundary layers and LES further from them. Boundary layers, or the area near the surface of your geometry, cause more complex turbulent behaviors. Theoretically, the fluid's velocity magnitude approaches 0 as you approach the geometry wall. Therefore, despite the accuracy trade off using RANS, you do not lose much practical information to your application. HRLES is not a singular model, it is a suite of different hybrid approaches that various commercial proprietary products such as ANSYS and OpenFOAM offer to clients. Note that in the commercial simulation software, Eddy Viscosity Models are implemented in addition to RANS and LES for the most robust calculations.

2.1.3 Finite Volume Method (FVM)

One of AB-UPT's core contributions is its ability to inference directly from the 3D CAD (computer aided design) geometry without requiring the volumetric meshing. This understated benefit requires additional FVM context to highlight it appropriately.

FVM is the dominant discretization technique for CFD. Since the Navier-Stokes equations solve fluid behaviors over a singular reference volume, the smaller the reference volume scale, the finer the calculation. AB-UPT references the HeXtreme algorithm [6] that was used in the DrivAerML dataset's simulations which represent high quality, industry standard, samples.

The DrivAerML dataset's paper [8] details the specific hyperparameters chosen for the HeXtreme algorithm. There are six hyperparameters involving boundary layer widths, number of boundary layers, and mesh density growth rates. Two additional mesh refinements were undertaken in the areas surrounding complex geometries and on the overall geometry to align the dataset to current industry meshing practices. In short, the meshing algorithm implementation is a complex and iterative process that requires substantial domain expertise. Once the hyperparameters are finetuned, the meshing algorithm alone contributes roughly 500 compute-hours to the $1M dataset.

2.2 Architecture Overview

AB-UPT is characterized by three core architectural features that work in synergy. This paper provides a summary of the features. Reference the AB-UPT paper for in-depth architectural details.

2.2.1 Multi-Branch Architecture

CFD simulations take geometry meshes and volume meshes as input. AB-UPT similarly splits its inputs into a surface and volume branch by separately encoding the 3D-coordinates of the meshes' nodes. However, AB-UPT uniquely incorporates the raw CAD geometry by sampling a 3D point cloud, unassociated with the nodes from the surface mesh, as a third branch of inputs.

Throughout the full architecture, cross attention shares information between branches, establishing contextual relationships between each branch's point cloud input. In the encoder, the geometry branch is cross-attended into the surface and volume branches before it is ultimately discontinued throughout the rest of the architecture. This grounds the surface and volumetric meshes in the geometry from which they were originally derived. This geometry context along with the encoded latent representations of the three branches effectively decouples the encoder from the decoder meaning, the outputs of the model are no longer specific to the exact locations of the surface and volume mesh representations.

The main takeaway from the multi-branch architecture is that, since the geometry branch is independent of any mesh structure, the model can now accept arbitrary geometries at inference.

2.2.2 Anchor Attention

Full self-attention is an $\mathcal{O}(n^2)$ operation where every token's query is multiplied by every token's key. Robust, industry standard, automotive CFD simulations, such as DrivAerML's, take roughly 140 million volumetric cells to calculate the level of fluid behavior detail necessary in practice. Full self-attention is not computationally feasible at that scale. The anchor attention feature reduces this scale by magnitudes (effective results at 4,000 anchors:140M queries) while maintaining the full volumetric mesh representation. Instead of full self-attention across all tokens, full self-attention occurs across a subsample of anchor tokens that are uniformly sampled from the surface and volume branches before the encoding block. The anchor tokens are responsible for making predictions at the rest of the tokens', or the query tokens, positions.

Not only does anchor attention make training with 140 million tokens tractable, but it also formulates the model output as a conditional neural field. Since the query tokens operate independently of one another, the anchor tokens' predictions are not constrained to the query tokens' coordinates and can make predictions at any relative position. Therefore, the model output is a continuous velocity vector field that is conditioned on the anchor tokens' predictions.

2.2.3 Hard Physics Constraints

Physics-informed neural networks, or PINNs, typically integrate physics formula residuals into the model's loss function. This encourages the model to produce real-world, physics consistent, outputs. This formulation is known as a soft-constraint because the physics-informed loss function does not guarantee physics-consistent predictions just as the model's loss can only approach 0.

AB-UPT enforces physics-consistency via a hard-constraint formulation. Hard-constraint formulations are significantly more challenging to implement because these formulas only work in the input domain, not the latent domain. The conditional neural field output byproduct of the anchor attention feature is a compatible domain of the Navier-Stokes equations. This enabled the authors to enforce the outputs to the divergence-free formulation ensuring no sinks or sources are present in the vorticity field.

3 Experiments

3.1 AB-UPT Benchmark

Per Table 1 AB-UPT outperforms all other existing methods with respect to accuracy and generalizability due to its mesh independent and neural field formulation. While Transformer's accuracies are comparable, predictions are mapped to specific mesh cell positions used as training input, limiting generalization to new meshes. One substantial improvement AB-UPT has over Transformer and Transolver is the vorticity error on the DrivAerML dataset due to its divergence-free hard constraint.

Table 1: Reproduction of Table 4 from AB-UPT: benchmarking against existing methods on DrivAerML.

	ShapeNet-Car		AhmedML			DrivAerML			Neural field	Mesh independent
	p_s	u	p_s	u	ω	p_s	u	ω		
PointNet	12.09	3.05	8.02	5.44	66.04	23.63	28.13	1747.7	✗	✗
GRAPH U-NET	10.33	2.49	6.46	4.15	53.66	16.13	17.98	540.6	✗	✗
GINO	13.28	2.53	7.90	6.23	71.81	13.03	40.58	131.7	✓	✓
LNO	9.05	2.29	12.95	7.59	72.49	20.51	23.27	493.8	✓	✓
UPT	6.41	1.49	4.25	2.73	15.03	7.44	8.74	90.2	✓	✓
OFormer	7.05	1.61	4.12	3.63	15.06	4.48	6.64	71.2	✓	✓
Transolver	6.46	1.62	3.45	2.05	8.22	4.81	6.78	38.4	✗	✗
Transformer	4.86	1.17	3.41	2.09	6.76	4.35	6.21	47.9	✗	✗
AB-UPT	**4.81**	**1.16**	**3.01**	**1.90**	**6.52**	**3.82**	**5.93**	**35.1**	✓	✓

Table 2: Reproduction of Table 3 from AB-UPT: ablation study isolating the contribution of each architectural feature.

		L2 error (↓)			Neural field	Mesh independent	Train speed	Test speed
		p_s	u	ω				
baseline	UPT	4.38	3.20	37.13	✓	✓	7.0	0.2
macro design	+ large decoder	4.25	2.73	15.03	✓	✓	14.7	1.4
	+ decoder-only with self-attn	3.41	2.09	6.76	✗	✗	14.9	163
re-add properties	+ anchor attention	3.41	2.09	6.76	✓	✗	14.9	2.6
	+ decouple geometry encoding	3.31	2.09	6.64	✓	✓	16.1	2.5
micro design	+ split surface/volume branch	3.50	2.02	6.91	✓	✓	11.0	1.5
	+ cross-branch interactions	3.35	2.08	6.76	✓	✓	11.0	1.5
	+ branch-specific decoders	3.01	1.90	6.52	✓	✓	11.0	1.5

Given that AB-UPT is already identified as the superior model architecture by accuracy, it makes Table 2's findings even more impactful. Table 2's ablation against the UPT baseline isolates each feature's accuracy contribution, but more importantly exposes its effect on train/test speed. Recall, the main bottleneck of CFD surrogate modeling is that the complexity of the volumetric meshing is an intractably large feature space for nearly all other models without an accuracy tradeoff. Anchor-attention elegantly addresses this problem. The model that only implements the "large decoder-only with self-attention" feature takes 163 seconds for inference. As soon as anchor-attention is added, that time is reduced to 2.6 seconds with the same accuracy.

Table 3: Reproduction of Table 5 from AB-UPT: CAD-only vs. volumetric mesh inference comparison.

	AhmedML			DrivAerML			Meshing	
	p_s	u	ω	p_s	u	ω	Surface mesh	Volume mesh
HRLES	0.00	0.00	0.00	0.00	0.00	0.0	CFD simulation mesh	CFD simulation mesh
HRLES	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	Geometry mesh (CAD)	Regular grid
AB-UPT	3.01	1.90	6.52	3.82	5.93	35.1	CFD simulation mesh	CFD simulation mesh
AB-UPT	3.70	2.50	7.59	4.14	6.64	51.2	Geometry mesh (CAD)	Regular grid

While there are numerous other experimentally proven benefits of AB-UPT's architecture in the original paper, the model's ability to inference with the CAD geometry alone without any prior meshing is both extremely practical and presents some incredibly interesting future work research that the paper does not address. The last row in Table 3 does show an accuracy trade-off between AB-UPT's inference with and without using the volumetric mesh, yet the loss without the volumetric mesh is comparable to the next best surrogate model. Avoiding the tedious finetuning, production time, and domain expertise requirements to create a volumetric mesh democratizes CFD simulations and improves generalizability greatly. In the future, I wonder if a similar Anchored-Branched approach can be taken to other physics simulations that use volumetric meshing to create a system of surrogate models conjoined by one latent geometry representation. Given physical constraints by an engineering team, I envision a parent model using this network of surrogates to automatically

optimize the geometry of the object before sending the surrogate optimal geometry to the final simulations for validation.

4 Related Works

The original AB-UPT paper's Related Works section references past works that cumulatively inspired and were leveraged in the AB-UPT architecture. In the Appendix section of AB-UPT, there are extended benchmarking tests against other, current, novel CFD surrogate architectures. I will briefly assess their methods and how they compare or contrast in architecture and ability to AB-UPT to contextualize AB-UPT to the broader CFD surrogate research space.

4.1 DoMINO [2]

DoMINO, now PhysicsNeMo, was introduced in January 2025 and is now packaged as an NVIDIA Inference Microservice. DoMINO's approach also involves point-clouds, but instead of using global-context through supernode pooling as AB-UPT does, it conducts a multi-scale, iterative local aggregation process. From the bottom-up, it continuously encodes larger and larger neighborhoods of point-clouds, scaling the local contextual information and reducing problem scale. Supernode pooling operations differ because each supernode contextualizes it's informational contribution to the entire point-cloud system. DoMINO was originally trained with DrivAerML and AB-UPT significantly outperformed it on accuracy benchmarks.

4.2 Erwin [5]

Erwin was introduced in June 2025. Erwin did not focus solely on CFD but did perform benchmarking on turbulent fluid dynamics datasets, specifically EAGLE and ShapeNet-Car. Erwin implements a "ball tree" architecture inspired by hierarchical tree-based algorithms and attention. Multihead attention is computed in parallel at fixed hierarchy levels of point-cloud neighborhoods. This hierarchical approach linearly scales the $\mathcal{O}(n^2)$ self-attention problem. AB-UPT notes this architecture as an interesting future direction they could extend anchor attention into to speed up computations. AB-UPT does outperform Erwin benchmarks on ShapeNet-Car but not significantly.

4.3 TripNet [3]

TripNet was introduced in October 2025. TripNet takes a fundamentally different approach to the other architectures. Instead of using point-clouds, it projects the raw geometry onto three orthogonal feature planes to engineer an organized continuous domain before using a U-Net and lightweight CNN to make the fluid flow predictions at those points. This approach allows a user to query specific sub-geometry fluid flow solutions. AB-UPT significantly outperforms TripNet on their common DrivAerNet++ benchmark.

4.4 Transolver++ [4]

Transolver++ was introduced in February 2025. Transolver [12] is part of the comprehensive benchmarking in the Results section of AB-UPT. Transolver++ implements sequence parallelism and is the more efficient version of Transolver. Transolver++ similarly compresses the volumetric mesh into a latent representation but does so by grouping point clouds into learned physical state tokens rather than supernode pooling. These physical states are learned aggregations of nodes that are associated with underlying physical behaviors of the solution space such as boundary layer turbulence, a wake, or a laminar region. The number of physical state tokens are directly controllable similarly to the number of anchor nodes in AB-UPT. Transolver++ was not directly reproducible and therefore not a part of the main benchmarking experiment. However, in their attempt to recreate Transolver++'s benchmarks, AB-UPT significantly outperformed them.

5 Summary

CFD is a highly complex, expensive, and necessary tool relied upon by critical engineering industries. A viable surrogate model to CFD would fundamentally change engineering workflows and their

efficiency. Through a synergy of novel architectural features, AB-UPT sets the standard of neural surrogate accuracy and inference time while guaranteeing physics consistent results and circumnavigating the FVM preprocessing step. AB-UPT's architecture may very well be the future of complex discretized physics simulation across multiple domains.

References

[1] Alkin, B., Bleeker, M., Kurle, R., Kronlachner, T., Sonnleitner, R., Dorfer, M., & Brandstetter, J. (2025) AB-UPT: Scaling Neural CFD Surrogates for High-Fidelity Automotive Aerodynamics Simulations via Anchored-Branched Universal Physics Transformers. *Transactions on Machine Learning Research* (10/2025). https://openreview.net/forum?id=nwQ8nitlTZ

[2] Ranade, R., Nabian, M. A., Tangsali, K., Kamenev, A., Hennigh, O., Cherukuri, R., & Choudhry, S. (2025) DoMINO: A Decomposable Multi-scale Iterative Neural Operator for Modeling Large Scale Engineering Simulations. *arXiv preprint arXiv:2501.13350*. https://arxiv.org/abs/2501.13350

[3] Chen, Q., Elrefaie, M., Dai, A., & Ahmed, F. (2025) TripNet: Learning Large-scale High-fidelity 3D Car Aerodynamics with Triplane Networks. *arXiv preprint arXiv:2503.17400*. https://arxiv.org/abs/2503.17400

[4] Luo, H., Wu, H., Zhou, H., Xing, L., Di, Y., Wang, J., & Long, M. (2025) Transolver++: An Accurate Neural Solver for PDEs on Million-Scale Geometries. *arXiv preprint arXiv:2502.02414*. https://arxiv.org/abs/2502.02414

[5] Zhdanov, M., Welling, M., & van de Meent, J.-W. (2025) Erwin: A Tree-based Hierarchical Transformer for Large-scale Physical Systems. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/2502.17019

[6] Jezdimirović, J., Chemin, A., Reberol, M., Henrotte, F., & Remacle, J.-F. (2021) Quad layouts with high valence singularities for flexible quad meshing. In *Proceedings of the 29th International Meshing Roundtable*. https://arxiv.org/abs/2103.02939

[7] Brunton, S. L. (2021, April 23) Turbulence Closure Models: Reynolds Averaged Navier Stokes (RANS) & Large Eddy Simulations (LES). *Cassyni seminar.* https://doi.org/10.52843/cassyni.cjkr7f

[8] Ashton, N., Mockett, C., Fuchs, M., Fliessbach, L., Hetmann, H., Knacke, T., Schonwald, N., Skaperdas, V., Fotiadis, G., Walle, A., Hupertz, B., & Maddix, D. (2024) DrivAerML: High-Fidelity Computational Fluid Dynamics Dataset for Road-Car External Aerodynamics. *arXiv preprint arXiv:2408.11969*. https://arxiv.org/abs/2408.11969

[9] Çengel, Y. A., & Cimbala, J. M. (2018) *Fluid Mechanics: Fundamentals and Applications*, 4th ed. New York: McGraw-Hill.

[10] Versteeg, H. K., & Malalasekera, W. (2007) *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd ed. Harlow: Pearson Education.

[11] Ardeshir, A. (n.d.) *Math 53 Lecture Notes, Section 16.5: Curl and Divergence.* University of California, Berkeley. https://math.berkeley.edu/~arash/53/notes/16_5.pdf

[12] Wu, H., Luo, H., Wang, H., Wang, J., & Long, M. (2024) Transolver: A Fast Transformer Solver for PDEs on General Geometries. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. https://arxiv.org/abs/2402.02366

A Navier-Stokes Equations Examples

A.1 Plane Couette Flow

This example applies the procedure from Çengel and Cimbala [9] to derive the velocity field for plane Couette flow from the incompressible Navier-Stokes equations. Plane Couette flow is the canonical shear-driven flow between two infinite parallel plates and serves as one of the few exact solutions to the Navier-Stokes equations.

Problem Setup. Consider steady, incompressible, fully developed laminar flow of a Newtonian fluid in the x-y plane between two infinite parallel plates separated by a distance h. The bottom plate at $y = 0$ is stationary and the top plate at $y = h$ moves in the $+x$ direction with constant velocity U. The fluid has density ρ and dynamic viscosity μ. See Figure 1.

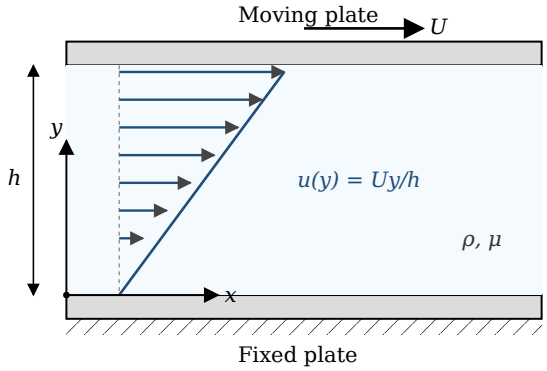

Figure 1: Plane Couette flow geometry. The top plate moves at velocity U in the $+x$ direction while the bottom plate is fixed. The fluid between the plates develops a linear velocity profile $u(y) = Uy/h$.

Assumptions and Boundary Conditions. The flow is assumed (i) steady: $\partial/\partial t = 0$; (ii) two-dimensional: $w = 0$, $\partial/\partial z = 0$; (iii) fully developed in x: $\partial u/\partial x = 0$; (iv) driven only by the top plate (no pressure gradient): $\partial p/\partial x = 0$; (v) gravity acts in $-y$ and is absorbed into a modified pressure. The no-slip boundary conditions are $u(0) = 0$ and $u(h) = U$.

Continuity. Under assumption (ii), the incompressible continuity equation (1) reduces to

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \tag{6}$$

Assumption (iii) gives $\partial u/\partial x = 0$, so $\partial v/\partial y = 0$. The no-penetration condition $v(0) = v(h) = 0$ forces $v \equiv 0$ throughout the domain.

Momentum. With $v = w = 0$, $\partial u/\partial x = 0$, and $\partial/\partial t = 0$, the x-momentum component of (2) reduces to

$$0 = -\frac{\partial p}{\partial x} + \mu \frac{\partial^2 u}{\partial y^2}. \tag{7}$$

Applying assumption (iv) eliminates the pressure gradient term:

$$\mu \frac{d^2 u}{dy^2} = 0. \tag{8}$$

The partial derivative becomes a total derivative since $u = u(y)$ only.

Integration. Integrating (8) twice yields

$$u(y) = C_1 y + C_2. \tag{9}$$

Boundary Conditions. Applying $u(0) = 0$ gives $C_2 = 0$. Applying $u(h) = U$ gives $C_1 = U/h$. The velocity field is therefore

$$u(y) = \frac{Uy}{h}. \tag{10}$$

Derived Quantities. The shear stress follows from Newton's law of viscosity:

$$\tau_{yx} = \mu \frac{du}{dy} = \frac{\mu U}{h}, \tag{11}$$

which is constant throughout the domain. The volumetric flow rate per unit depth is

$$Q = \int_0^h u(y)\, dy = \frac{Uh}{2}. \tag{12}$$

Verification. Substituting (10) into (8) gives $d^2u/dy^2 = 0$, satisfying the reduced momentum equation. Both boundary conditions are satisfied by inspection. The linear profile confirms that momentum is transferred by pure viscous diffusion in the absence of pressure or inertial forcing.

B FVM Solve Example

This example demonstrates the finite volume discretization process referenced in Section 2.1 on a simplified version of Example 4.1 from Versteeg and Malalasekera [10]. Their original example uses five control volumes to discretize 1D steady-state heat conduction through an insulated rod; this appendix reduces the grid to two control volumes to illustrate the flux-balance mechanics with minimal notation. The same discretization logic extends directly to the viscous diffusion term $\mu\nabla^2 u$ in the Navier-Stokes equations.

B.1 Governing Equation and Problem Setup

Following [10], consider source-free, steady-state heat conduction in an insulated rod with ends held at constant temperatures $T_A = 100°\text{C}$ and $T_B = 500°\text{C}$. The governing equation is

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right) = 0, \tag{13}$$

with $L = 0.5$ m, $k = 1000$ W/m·K, and cross-sectional area $A = 10 \times 10^{-3}$ m^2.

B.2 Discretization

The rod is divided into two equal control volumes with nodes placed at the cell centers and boundary nodes placed at the rod ends (Figure 2). This gives a control-volume width of $\Delta x = L/2 = 0.25$ m and a node spacing of $\delta x = L/2 = 0.25$ m between interior nodes, with $\delta x/2 = 0.125$ m between each boundary node and its adjacent interior node.

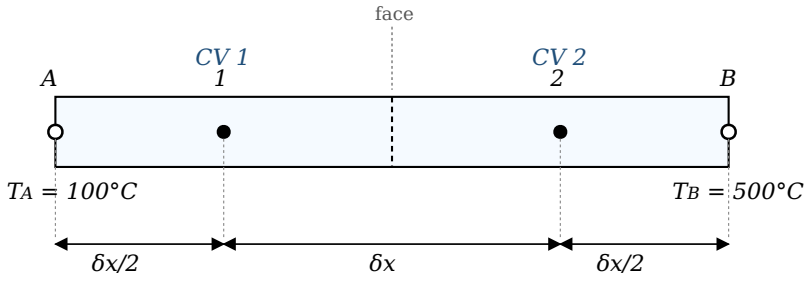

Figure 2: Two control volume discretization of the insulated rod. Cell centers are labeled 1 and 2, boundary nodes are A and B, and cell faces are marked with dashed lines.

B.3 Flux Balance

Integrating (13) over a control volume and applying the divergence theorem yields the conservative flux balance

$$\left(kA\frac{dT}{dx}\right)_e - \left(kA\frac{dT}{dx}\right)_w = 0, \tag{14}$$

where subscripts e and w denote the east and west faces of the control volume. The face gradients are approximated by central differences between adjacent nodal temperatures.

Control volume 1. The west face coincides with boundary A (distance $\delta x/2$ from node 1) and the east face lies between nodes 1 and 2 (distance δx apart):

$$kA\frac{T_2 - T_1}{\delta x} - kA\frac{T_1 - T_A}{\delta x/2} = 0. \tag{15}$$

Control volume 2. By symmetry:

$$kA\frac{T_B - T_2}{\delta x/2} - kA\frac{T_2 - T_1}{\delta x} = 0. \tag{16}$$

B.4 Discretized System

Dividing through by kA and collecting terms yields the linear system

$$3T_1 - T_2 = 2T_A, \tag{17}$$
$$-T_1 + 3T_2 = 2T_B. \tag{18}$$

Substituting $T_A = 100$ and $T_B = 500$:

$$3T_1 - T_2 = 200, \tag{19}$$
$$-T_1 + 3T_2 = 1000. \tag{20}$$

B.5 Solution

Solving the 2×2 system gives

$$T_1 = 200^\circ\mathrm{C}, \qquad T_2 = 400^\circ\mathrm{C}. \tag{21}$$

B.6 Verification

The analytical solution to (13) with the given boundary conditions is the linear profile $T(x) = T_A + (T_B - T_A)x/L$. Evaluating at the cell-center positions $x_1 = 0.125$ m and $x_2 = 0.375$ m gives $T(x_1) = 200^\circ$C and $T(x_2) = 400^\circ$C, which match (21) exactly. Agreement is expected since central differencing is second-order accurate and reproduces linear solutions without truncation error.

C Divergence-Free Formulation Example

AB-UPT enforces the divergence-free vorticity condition $\nabla \cdot \boldsymbol{\omega} = 0$ as an architectural hard constraint by defining the predicted vorticity as the curl of the predicted velocity, $\hat{\boldsymbol{\omega}}(\mathbf{x}) = \nabla \times \hat{\mathbf{u}}(\mathbf{x})$. This guarantees $\nabla \cdot \hat{\boldsymbol{\omega}} = 0$ by construction via the standard vector calculus identity that the divergence of a curl vanishes identically. This appendix reproduces the derivation of the identity following Ardeshir's lecture notes for Stewart's Section 16.5 [11].

C.1 Curl

Define the del operator

$$\nabla = \left\langle \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right\rangle. \tag{22}$$

The curl of a vector field $\mathbf{F} = \langle P, Q, R \rangle$ is defined as the cross product of ∇ with $\mathbf{F}$:

$$\operatorname{curl}\mathbf{F} = \nabla \times \mathbf{F} = \begin{vmatrix} \hat{\imath} & \hat{\jmath} & \hat{k} \\ \partial/\partial x & \partial/\partial y & \partial/\partial z \\ P & Q & R \end{vmatrix} = \left(\frac{\partial R}{\partial y} - \frac{\partial Q}{\partial z}\right)\hat{\imath} + \left(\frac{\partial P}{\partial z} - \frac{\partial R}{\partial x}\right)\hat{\jmath} + \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right)\hat{k}. \tag{23}$$

C.2 Divergence

If $\mathbf{F} = \langle P, Q, R \rangle$ is a vector field on $\mathbb{R}^3$ and $\partial P/\partial x$, $\partial Q/\partial y$, and $\partial R/\partial z$ exist, then the divergence of $\mathbf{F}$ is

$$\operatorname{div}\mathbf{F} = \nabla \cdot \mathbf{F} = \frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y} + \frac{\partial R}{\partial z}. \tag{24}$$

Note that $\operatorname{curl}\mathbf{F}$ is a vector field while $\operatorname{div}\mathbf{F}$ is a scalar field.

C.3 Divergence of a Curl

Theorem. If $\mathbf{F} = \langle P, Q, R \rangle$ is a vector field on $\mathbb{R}^3$ and P, Q, R have continuous second-order partial derivatives, then

$$\operatorname{div} \operatorname{curl} \mathbf{F} = 0. \tag{25}$$

Proof. Applying the definitions in (23) and (24):

$$\begin{aligned} \operatorname{div} \operatorname{curl} \mathbf{F} = \nabla \cdot (\nabla \times \mathbf{F}) &= \frac{\partial}{\partial x}\left(\frac{\partial R}{\partial y} - \frac{\partial Q}{\partial z}\right) + \frac{\partial}{\partial y}\left(\frac{\partial P}{\partial z} - \frac{\partial R}{\partial x}\right) + \frac{\partial}{\partial z}\left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right) \\ &= \frac{\partial^2 R}{\partial x \partial y} - \frac{\partial^2 Q}{\partial x \partial z} + \frac{\partial^2 P}{\partial y \partial z} - \frac{\partial^2 R}{\partial y \partial x} + \frac{\partial^2 Q}{\partial z \partial x} - \frac{\partial^2 P}{\partial z \partial y} \\ &= 0, \end{aligned} \tag{26}$$

because the terms cancel in pairs by Clairaut's Theorem (equality of mixed partials). This identity holds pointwise for any C^2 vector field, independent of boundary conditions or flow regime.

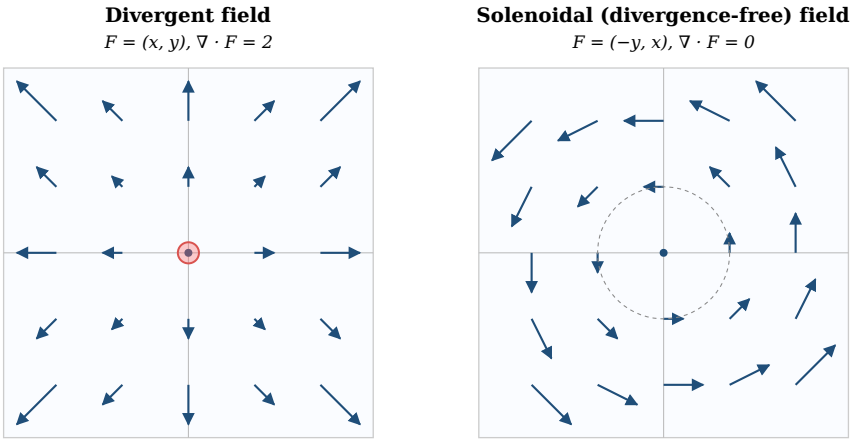

Figure 3: Left: a divergent 2D vector field $\mathbf{F} = (x, y)$ with sources radiating outward ($\nabla \cdot \mathbf{F} = 2 \neq 0$). Right: a solenoidal 2D vector field $\mathbf{F} = (-y, x)$ with closed circulation and no net outflow ($\nabla \cdot \mathbf{F} = 0$). The vorticity field in AB-UPT is constrained to the solenoidal class by construction.

D Related Works Architectures

This appendix provides architectural reference diagrams for the four comparable CFD surrogate methods discussed in Section 4. Each figure is reproduced from its respective paper and illustrates the core architectural mechanism that contrasts with AB-UPT's multi-branch + anchor-attention approach.

D.1 DoMINO

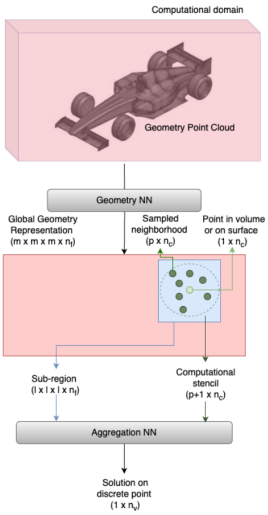

Figure D.1: DoMINO architecture [2]. In contrast to AB-UPT's supernode pooling, which provides global context by allowing each supernode to aggregate information across the full point cloud, DoMINO performs multi-scale iterative local aggregation. Encoding proceeds bottom-up through progressively larger neighborhoods of points, reducing problem scale by compounding local context rather than establishing global context in a single pooling step.

D.2 Erwin

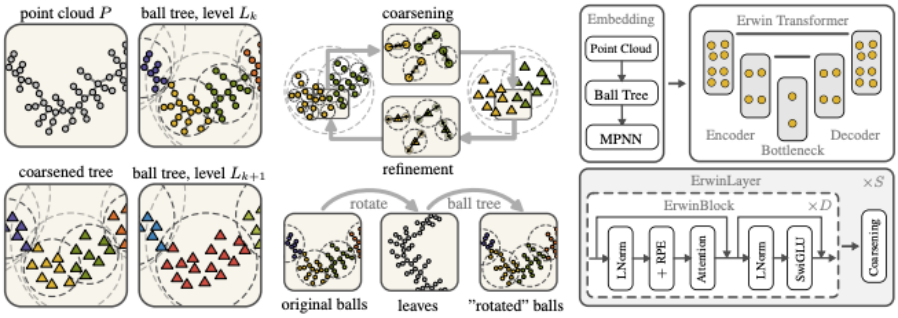

Figure D.2: Erwin architecture [5]. Erwin implements a hierarchical "ball tree" structure that linearly scales self-attention. Multihead attention is computed in parallel at fixed hierarchy levels of point-cloud neighborhoods rather than across the full token set. This is architecturally distinct from AB-UPT's anchor attention, which reduces complexity by compressing the attention computation onto a small subsampled anchor set while preserving the full query resolution via cross-attention.

D.3 TripNet

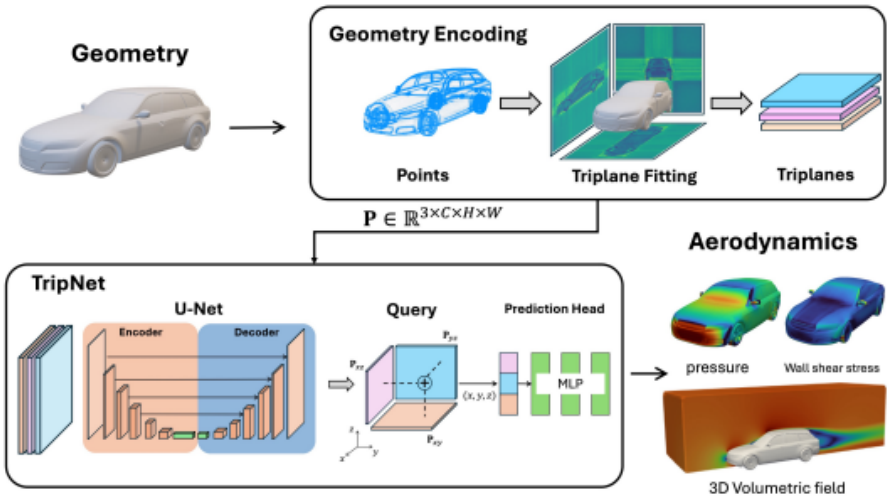

Figure D.3: TripNet architecture [3]. TripNet abandons point-cloud representations entirely and instead projects the raw geometry onto three orthogonal feature planes, producing an organized continuous domain that is processed by a U-Net and a lightweight CNN. This triplane representation enables querying fluid flow solutions over specific sub-geometries, contrasting with AB-UPT's unified point-cloud branches that decouple geometry, surface, and volume meshes.

D.4 Transolver++

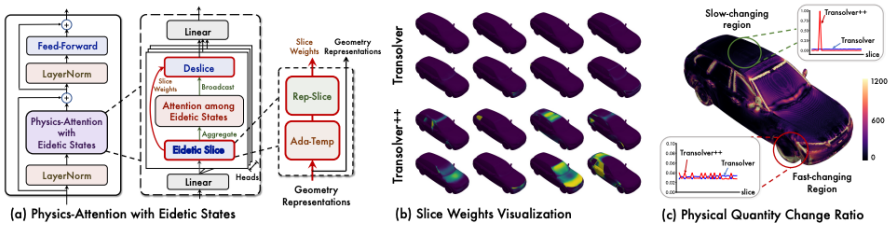

Figure D.4: Transolver++ architecture [4]. Transolver++ compresses the volumetric mesh into a latent representation by grouping point clouds into learned physical state tokens that correspond to underlying flow behaviors (e.g., boundary layer turbulence, wake regions, laminar regions). The number of physical state tokens is directly controllable, analogous to the number of anchor tokens in AB-UPT, but the tokens are learned semantic aggregations rather than a subsampled subset of mesh positions. Transolver++ additionally implements sequence parallelism for efficiency over its predecessor Transolver [12].

E AB-UPT Tutorial Output

This appendix provides the outputs from Emmi AI's AB-UPT tutorial where you can visualize inference examples from the trained AB-UPT model. See AB-UPT Tutorial.

E.1 Raw CAD Geometry

Figure E.1: Raw CAD Geometry. Due to AB-UPT's multi-branch architecture, this is an example raw CAD geometry that can be used as sole input to the model during inference.

E.2 Surface Pressure Prediction

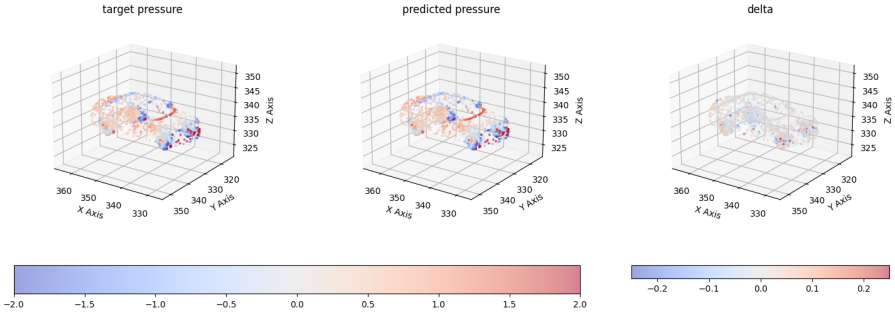

Figure E.2: Surface Pressure Prediction. The "target pressure" plot [left] shows the ground truth surface pressure values. The "predicted pressure" plot [center] shows the predicted surface pressure values. The "delta" plot [right] shows the difference between point wise values of the ground truth and predicted values. Visually, the greater deltas appear to conglomerate around complex geometries of the car such as the wheels and the geometry's surface transitions.

E.3 X-Velocity Prediction

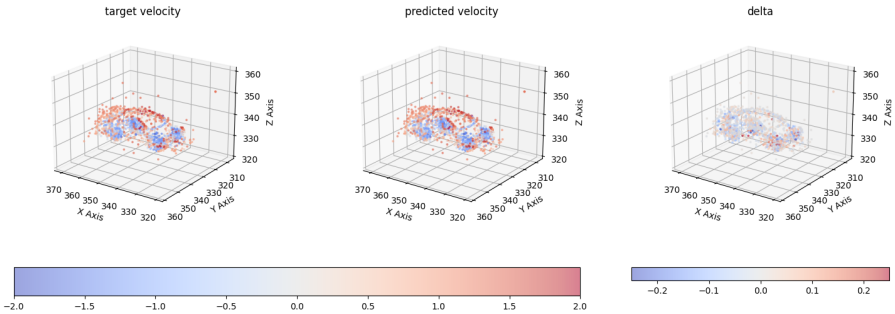

Figure E.3: X-Velocity Prediction. The "target velocity" plot [left] shows the ground truth x-velocity values. The "predicted velocity" plot [center] shows the predicted x-velocity values. The "delta"

plot [right] shows the difference between point wise values of the ground truth and predicted values. Visually, the greater deltas also appear to conglomerate around complex geometries of the car such as the wheels and the geometry's surface transitions. The deltas appear to decrease in magnitude as you look at nodes further away from the surface of the geometry.

E.4 Streamline Comparison

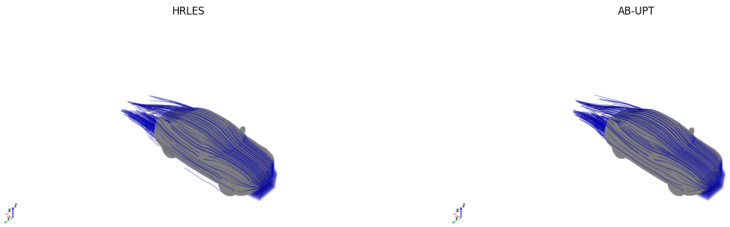

Figure E.4: Streamline Comparison. The "HRLES" plot [left] shows the ground truth [x,y,z] vectors. Recall HRLES is the industry accepted turbulence model from which DrivAerML simulations were derived with. The "AB-UPT" plot [right] shows the predicted [x,y,z] vectors of the same geometry. There is no discernible difference between the two at this scale and resolution.