DAFL: DISTURBANCE-AWARE FIELD LEARNING FOR ROBUST CFD WARM-STARTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Aerodynamic design optimization necessitates repeated analyses across diverse geometries and flow conditions, each requiring costly computational fluid dynamics (CFD) simulations. Learning-based warm-starting offers a promising way to reduce solver iterations by predicting initial states closer to convergence. However, existing methods can fail on complex three-dimensional aerodynamic flows, where spatially heterogeneous disturbances and highly imbalanced flow variable distributions lead to inaccurate or even invalid predictions that undermine solver convergence. We propose Disturbance-aware Field Learning (DAFL), a framework for robust CFD warm-starting. DAFL adaptively partitions the flow field according to disturbance characteristics and performs disturbance-aware sampling and regression, enabling accurate modeling of critical disturbances. Geometry and flow conditions are injected throughout the backbone to capture case-dependent variations, while logarithmic wall distance improves the representation of steep near-wall velocity gradients. For flow variables with highly imbalanced distributions, DAFL performs regression in log space and maps predictions back through an exponential transformation, improving optimization while preserving physically valid initial states. Experiments on complex three-dimensional aircraft data demonstrate that DAFL consistently outperforms existing approaches in flow field prediction. More importantly, DAFL enables robust CFD warm-starting, substantial solver acceleration, and strong generalization across diverse geometries and flow conditions.

1 INTRODUCTION

Computational fluid dynamics (CFD) is widely used to evaluate aircraft aerodynamic performance, but resolving complex three-dimensional (3D) flows remains computationally expensive. Learning-based methods can reduce this cost either by directly approximating flow solutions or by assisting numerical solvers during iteration. Direct surrogate methods treat predicted flow fields as final solutions without numerical iterations or convergence checks (Guo et al., 2016; Bhatnagar et al., 2019; Thuerey et al., 2020; Li et al., 2021; Lu et al., 2021; Bonnet et al., 2022). In contrast, solver-coupled methods incorporate learned components into the numerical iteration process (Obiols-Sales et al., 2020; Um et al., 2020; Kochkov et al., 2021; Li et al., 2024; Zhang et al., 2024). Warm-starting is one such solver-coupled strategy that replaces the uniform initial state used in cold-starting with a predicted initial state (Figure 1). When the predicted initial state provides a sufficiently accurate approximation of the converged solution, warm-starting can reduce solver iterations. However, complex 3D aerodynamic flows present two challenges that can produce inaccurate or invalid initial states and undermine solver convergence.

The first challenge arises from spatially heterogeneous disturbances across the flow field. Large weak-disturbance regions remain close to the inflow state, whereas strong-disturbance regions are localized near walls and in wakes. The locations and extents of these regions vary with geometry and flow conditions. Near-wall predictions are particularly difficult because steep velocity gradients occur over a narrow wall-normal range. A uniform processing strategy may therefore fail to capture these critical disturbances accurately.

The second challenge is the highly imbalanced distribution of flow variables, particularly turbulence variables (Appendix C). These variables span a wide range, with low values occurring frequently and

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.

high values only rarely. This imbalance makes direct physical-space regression more difficult and can reduce regression accuracy. Predictions from direct physical-space regression may also violate the positivity constraints for density, pressure, and turbulence variables. Such violations produce physically invalid initial states and can undermine solver convergence.

To address these challenges, we propose Disturbance-aware Field Learning (DAFL) for robust CFD warm-starting. DAFL handles spatially heterogeneous disturbances by adaptively partitioning the flow field into weak-disturbance, near-wall strong-disturbance, and outer strong-disturbance regions. Based on this adaptive partition, DAFL performs disturbance-aware sampling and regression to model critical disturbances accurately. Geometry and flow conditions are injected throughout the backbone to capture case-dependent variations. Logarithmic wall distance improves the representation of steep near-wall velocity gradients. To address highly imbalanced flow variable distributions, DAFL performs regression in log space and maps the predictions back through an exponential transformation. This transformation produces physically valid initial states for CFD warm-starting.

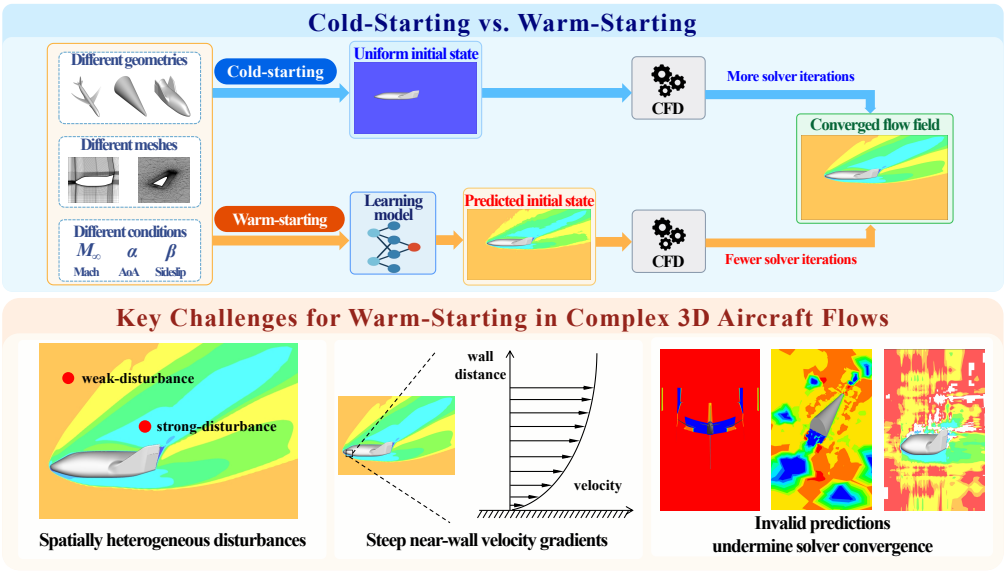


Figure 1: CFD cold-starting vs. learning-based warm-starting and key challenges in complex 3D aircraft flows.

Contributions.

- We propose DAFL, a framework that explicitly addresses spatially heterogeneous disturbances, case-dependent variations, and steep near-wall velocity gradients to improve the modeling of critical disturbances.
- For highly imbalanced flow variable distributions, DAFL performs regression in log space to improve optimization and applies an exponential transformation to produce physically valid initial states for CFD warm-starting.
- Experiments on complex 3D aircraft data show that DAFL consistently outperforms existing approaches in flow field prediction and enables robust CFD warm-starting with substantial solver acceleration. Interpolation and extrapolation tests further demonstrate strong generalization across diverse geometries and flow conditions.

2 Related Work

Learning-based flow field prediction. Early convolutional models for flow field prediction rely on regular spatial inputs (Guo et al., 2016; Bhatnagar et al., 2019; Thuerey et al., 2020). To handle irregular spatial layouts, point cloud models directly process irregular points (Xue et al., 2026; Hwang et al., 2025; Gu et al., 2026), while graph neural networks capture interactions among points or mesh

cells (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021; Mousavi et al., 2025). Existing models further consider varying flow conditions and geometries. Point cloud models incorporate flow conditions for surface pressure prediction (Xue et al., 2026; Zou et al., 2026). Neural operators learn mappings between function spaces (Li et al., 2021; Lu et al., 2021), and incorporating geometric information extends these mappings to different geometries (Li et al., 2023a;b; Serrano et al., 2023; Wen et al., 2025). Architectures including LNO, Transolver-3, and AB-UPT improve computational efficiency for large-scale meshes (Wang & Wang, 2024; Zhou et al., 2026; Alkin et al., 2025). These models enable flow prediction under varying conditions, complex geometries, and large-scale meshes.

Modeling spatially heterogeneous disturbances. Transolver groups mesh points with similar physical states into learned slices (Wu et al., 2024a). GNOT uses geometric gating for soft domain decomposition (Hao et al., 2023), whereas neural operators using domain decomposition combine local solutions across subdomains (Huang et al., 2026). However, these strategies do not explicitly partition each flow field according to its actual disturbance characteristics. This limitation is particularly pronounced in complex 3D aerodynamic flows. Large portions of the domain remain close to the inflow state, whereas strong disturbances are localized near walls and in wakes. The locations and extents of these strongly disturbed regions vary with geometry and flow conditions. Near-wall prediction is further complicated by steep velocity gradients within a narrow wall-normal region.

Predicting initial states for CFD warm-starting. Unlike direct surrogates, which treat predicted flow fields as final solutions, learning-based warm-starting uses these predictions to initialize subsequent CFD iterations. This approach retains the numerical solution process while reducing the number of iterations required for convergence (Chen et al., 2024; Xiao et al., 2024). Zhang et al. (2024) initialize CFD simulations with flow fields predicted by a residual graph convolutional network. Stumpo et al. (2026) use graph neural networks to predict initial states for hypersonic flow simulations. These studies demonstrate the potential of learning-based warm-starting to reduce solver iterations. However, inaccurate or invalid initial states can undermine solver convergence in complex 3D aerodynamic flows, particularly when flow variable distributions are highly imbalanced.

3 METHOD

3.1 PROBLEM SETUP AND FRAMEWORK OVERVIEW

For each CFD case, the inputs to DAFL consist of an aircraft surface geometry $\mathcal{S}$, a computational mesh represented by its N cell centers $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, and flow conditions $\mathbf{c} = (M_\infty, \alpha, \beta)$. Here, M_∞, α, and β denote the inflow Mach number, angle of attack, and angle of sideslip, respectively. For each cell center $\mathbf{x}_i$, the wall distance is computed as $d_i = \mathrm{dist}(\mathbf{x}_i, \mathcal{S}) > 0$. The corresponding converged CFD field is denoted by $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$, where $\mathbf{y}_i$ is the vector of flow variables at $\mathbf{x}_i$. Given these inputs, DAFL predicts an initial flow field $\hat{\mathbf{Y}}$ for CFD warm-starting.

As shown in Figure 2, DAFL first adaptively partitions the flow field according to disturbance characteristics and then uses the resulting partition to guide disturbance-aware sampling and regression. The disturbance partition prediction and full field regressor use a Point Transformer V3 (PTv3) backbone (Wu et al., 2024b). Geometry and flow conditions are injected throughout the backbones, and logarithmic wall distance is used as a pointwise input feature. Positive flow variables are regressed in log space and mapped back to their original physical scale through an exponential transformation. Finally, the resulting field $\hat{\mathbf{Y}}$ is used as the initial state for the CFD solver.

3.2 DISTURBANCE-AWARE FIELD LEARNING

Disturbance partition prediction. For a point with local velocity $\mathbf{v} = (u, v, w)$, the velocity disturbance relative to the inflow velocity $\mathbf{v}_\infty = (u_\infty, v_\infty, w_\infty)$ is defined as

$$\Delta\mathbf{v} = \mathbf{v} - \mathbf{v}_\infty = (u - u_\infty,\, v - v_\infty,\, w - w_\infty)\,. \tag{1}$$

We construct binary disturbance labels from $\Delta\mathbf{v}$ using Gaussian mixture models, as described in Algorithm 1. The binary labels $b_i^\star = 0$ and $b_i^\star = 1$ indicate points with weak and strong disturbances, respectively.

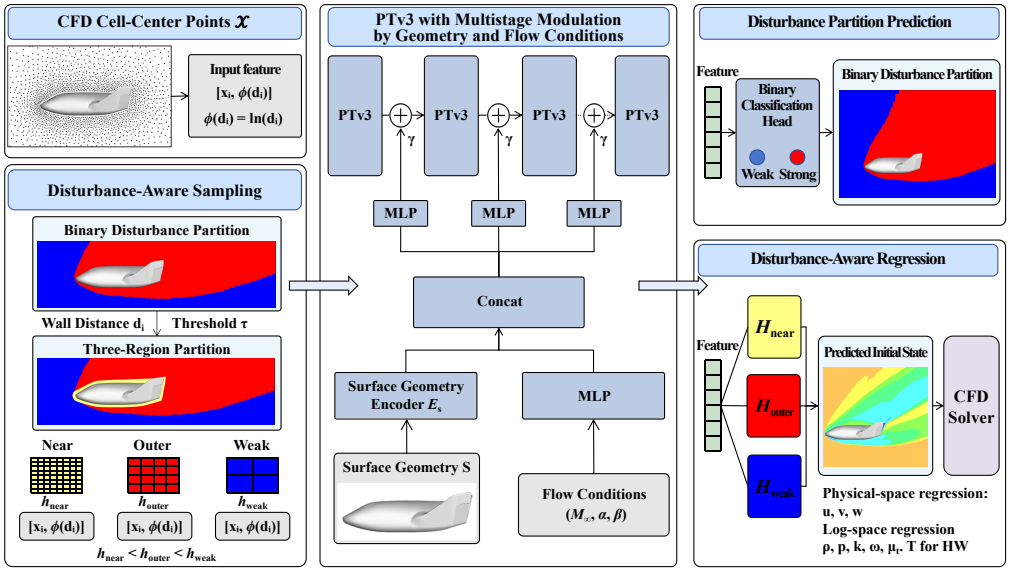


Figure 2: DAFL framework for CFD warm-starting. DAFL adaptively partitions the flow field to guide disturbance-aware sampling and regression, integrates geometry and flow conditions through multistage modulation, and employs features based on logarithmic wall distance and log-space regression for positive flow variables.

Since the ground-truth disturbance labels are unavailable during inference, DAFL includes a binary disturbance partition predictor. As shown in Figure 2, the central module uses a PTv3 backbone G_Φ to extract features. Geometry and flow conditions are injected throughout this backbone. The binary classification head H_{cls} maps each feature to probabilities over the two disturbance classes:

$$\mathbf{f}_i^{(\text{cls})} = G_\Phi\left(\mathbf{x}_i, \phi(d_i), \mathbf{g}\right), \qquad \mathbf{p}_i = \text{softmax}\Big(H_{\text{cls}}\big(\mathbf{f}_i^{(\text{cls})}\big)\Big), \qquad \widehat{b}_i = \underset{k\in\{0,1\}}{\arg\max}\, p_{i,k}. \tag{2}$$

Here, $\mathbf{f}_i^{(\text{cls})}$ denotes the feature vector produced by the classification backbone G_Φ. The probability vector $\mathbf{p}_i = (p_{i,0}, p_{i,1})$ contains the predicted probabilities of weak and strong disturbances, respectively. The representation $\mathbf{g}$ of geometry and flow conditions and the feature $\phi(d_i)$ based on logarithmic wall distance are defined in Sections 3.3 and 3.4, respectively.

To account for steep near-wall velocity gradients, we further divide the predicted strong-disturbance region according to wall distance. Given a threshold τ, each point is assigned to one of three regions:

$$\widehat{r}_i = \begin{cases} \text{weak}, & \widehat{b}_i = 0, \\ \text{near}, & \widehat{b}_i = 1,\ d_i < \tau, \\ \text{outer}, & \widehat{b}_i = 1,\ d_i \geq \tau. \end{cases} \tag{3}$$

We set a fixed threshold τ for each dataset. The threshold values are reported in Appendix Table 3.

Disturbance-aware sampling. To model critical disturbances accurately while reducing training cost, DAFL adapts the sampling resolution to disturbance characteristics by voxelizing each predicted region separately with its own voxel size h_r:

$$\mathbf{q}_i^{(r)} = \left\lfloor \frac{\mathbf{x}_i}{h_r} \right\rfloor, \qquad h_{\text{near}} < h_{\text{outer}} < h_{\text{weak}}. \tag{4}$$

Here, $\mathbf{q}_i^{(r)}$ denotes the voxel index of point $\mathbf{x}_i$ in region r, with the floor operation applied componentwise. This ordering assigns the finest sampling resolution to the near-wall region, followed by the outer and weak-disturbance regions. At each training iteration, one mesh point is randomly selected from every nonempty voxel. This sampling scheme provides more uniform spatial coverage than direct random sampling.

Disturbance-aware regression. The full field regressor adapts to disturbance characteristics through three lightweight regression heads, one for each region. All regions share a PTv3 backbone G_Θ:

$$\mathbf{f}_i^{(\mathrm{reg})} = G_\Theta\left(\mathbf{x}_i, \phi(d_i), \mathbf{g}\right), \qquad \widehat{\mathbf{y}}_i = H_{\widehat{r}_i}\left(\mathbf{f}_i^{(\mathrm{reg})}\right). \tag{5}$$

Here, $\mathbf{f}_i^{(\mathrm{reg})}$ denotes the feature vector produced by the shared regression backbone G_Θ. The predicted region label $\widehat{r}_i$ selects the corresponding regression head $H_{\widehat{r}_i}$, which maps this feature to the regression output $\widehat{\mathbf{y}}_i$. The regression and classification backbones have the same architecture.

3.3 Multistage Modulation by Geometry and Flow Conditions

The locations and extents of spatially heterogeneous disturbances vary across diverse geometries and flow conditions. To capture these case-dependent variations, we encode the surface geometry $\mathcal{S}$ and flow conditions $\mathbf{c}$ separately and concatenate their representations:

$$\mathbf{g}_\mathrm{s} = E_\mathrm{s}(\mathcal{S}), \qquad \mathbf{g}_\mathrm{c} = \mathrm{MLP}_\mathrm{cond}(\mathbf{c}), \qquad \mathbf{g} = \mathbf{g}_\mathrm{s} \parallel \mathbf{g}_\mathrm{c}. \tag{6}$$

Here, E_s is the encoder for surface geometry, and $\mathrm{MLP}_\mathrm{cond}$ is a multilayer perceptron that encodes the flow conditions. The operator $\parallel$ denotes feature concatenation. Additional implementation details are provided in Appendix B.3.

Geometry and flow conditions are injected into every stage of backbones through the representation $\mathbf{g}$. At stage ℓ, the point features $\mathbf{F}_\ell$ are modulated as

$$\mathbf{F}'_\ell = \mathbf{F}_\ell + \gamma_\ell \,\mathrm{MLP}_\ell(\mathbf{g}). \tag{7}$$

Here, MLP_ℓ maps $\mathbf{g}$ to the feature dimension of $\mathbf{F}_\ell$. The learnable scaling factor γ_ℓ controls the contribution of the modulation term.

3.4 Logarithmic Representation of Wall Distance

Steep near-wall velocity gradients are concentrated within a narrow wall-normal range. To represent these gradients more effectively, we use the natural logarithm of wall distance:

$$\phi(d_i) = \ln(d_i), \qquad d_i > 0. \tag{8}$$

The logarithmic transformation increases the separation among small wall distances while compressing the variation among larger distances. This effect is illustrated in Appendix Figure 10. We concatenate $\phi(d_i)$ with the point coordinates $\mathbf{x}_i$ to form the input features.

3.5 Log-Space Regression for Imbalanced Flow Variables

Highly imbalanced distributions of flow variables contain many low values and relatively few high values, particularly for turbulence variables (Appendix C). Direct regression on the original physical scale can therefore be difficult. We instead use the natural logarithm to define the regression targets and map predictions back through an exponential transformation:

$$z = \ln y, \qquad \widehat{y} = \exp(\widehat{z}) > 0. \tag{9}$$

Here, z and $\widehat{z}$ are the target and prediction in log space, respectively. The logarithmic transformation improves optimization, while the exponential transformation enforces positivity and preserves physically valid initial states.

4 Experiments

We evaluate DAFL in terms of flow field prediction, CFD warm-starting, and solver acceleration across diverse geometries and flow conditions. We further conduct ablation studies to assess the contributions of disturbance-aware field learning and log-space regression to prediction accuracy and the physical validity of predicted initial states.

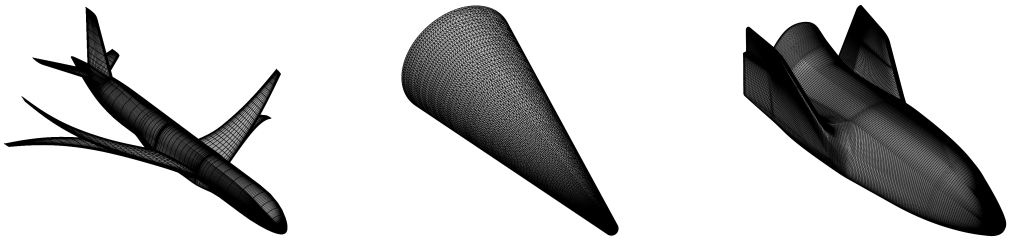
4.1 EXPERIMENTAL SETUP

Datasets and splits. Figure 3 presents the geometries and computational meshes for the three complex 3D aircraft datasets. These datasets evaluate DAFL's generalization across diverse geometries and flow conditions, as well as its applicability to structured and unstructured meshes. All datasets use meshes with millions of cells, with X38 being the largest at 7.41 million cells. For evaluation, each dataset includes six test cases covering both interpolation and extrapolation. Detailed training configurations and test splits are provided in Appendix A.

CHN-T2 (Liu et al., 2023) is a wide-body aircraft configuration with supercritical wings. It tests generalization across diverse aircraft geometries with varying wing deflections under a fixed flow condition. The training set comprises 24 geometries with upward or downward wing deflections at different strength levels. Each geometry uses a structured mesh of 1,091,432 cells.

HW (Kimmel et al., 2015; Willems et al., 2014) is a conical warhead configuration designed for hypersonic flight. It tests generalization across flow conditions on a fixed geometry. The training set comprises 32 flow conditions with different combinations of inflow Mach number, angle of attack, and angle of sideslip. Each case uses an unstructured mesh of 1,645,295 cells.

X38 (Loomis et al., 1997; Horvath et al., 2004) is a lifting-body configuration developed for NASA's X-38 Crew Return Vehicle program. It tests generalization across angles of attack on a fixed geometry. The training set comprises 19 flow conditions with different angles of attack. Each case uses a structured mesh of 7,412,224 cells.



CHN-T2, structured mesh | HW, unstructured mesh | X38, structured mesh

Figure 3: Geometries and computational meshes for CHN-T2, HW, and X38. CHN-T2 varies wing deflection under fixed flow conditions, whereas HW and X38 vary flow conditions for fixed geometries.

Baselines. We compare DAFL with five baselines: LNO (Wang & Wang, 2024), GINO (Li et al., 2023b), Transolver-3 (Zhou et al., 2026), AB-UPT (Alkin et al., 2025), and AdaField (Zou et al., 2026). We adapt these baselines to predict the complete set of flow variables required for CFD warm-starting (Appendix B.4). The targets for CHN-T2 and X38 are density ρ, velocity components (u, v, w), static pressure p, turbulent kinetic energy k, specific dissipation rate ω, and eddy viscosity μ_t. HW additionally includes temperature T. Predicting these variables is challenging due to spatially heterogeneous disturbances and highly imbalanced distributions, particularly for turbulence variables (Appendix C).

Metrics. The disturbance partition predictor identifies the regions used for subsequent field prediction. As an intermediate step, its classification performance is reported in Appendix C.2. The main evaluation focuses on full field prediction and CFD warm-starting. We assess full field accuracy using relative L_2 error (RL2) and report mean squared error (MSE) and mean absolute error (MAE) for each variable. Warm-starting uses a predicted initial state and cold-starting uses a uniform one, with identical solver settings, iteration limits, and convergence criteria. We report the number of converged cases and the iteration speedup. Further details are provided in Appendices A, D, and E.

4.2 FLOW FIELD PREDICTION

Accuracy and generalization. DAFL achieved the lowest RL2 across all three datasets (Table 1). Compared with the best performing baseline, it reduced the error by 27%, 57%, and 74% on CHN-T2, HW, and X38, respectively. DAFL uses only 16.72M parameters, 11%–45% fewer than the

Table 1: Full field RL2 and parameter counts of the flow field prediction models. Bold and underlined values denote the best overall result and the best baseline, respectively.

Method	Parameters (M)	CHN-T2	HW	X38
LNO	20.02	1.96	1.38	1.05
GINO	27.00	1.04	0.988	0.773
Transolver-3	21.85	3.03	1.10	0.887
AB-UPT	18.76	<u>0.494</u>	0.617	<u>0.476</u>
AdaField	30.61	0.529	<u>0.470</u>	0.693
DAFL	16.72	**0.359**	**0.201**	**0.125**

baselines. Results for individual test cases further demonstrate that DAFL maintains strong predictive performance across diverse geometries and flow conditions under both interpolation and extrapolation settings (Appendix Table 8). At the variable level, DAFL also generally outperforms the baselines (Appendix Table 9).

Reconstruction of critical disturbances. Figure 4 compares Mach number and turbulent kinetic energy (k) fields. Mach number is sensitive to velocity, pressure, and density prediction errors and may become undefined when pressure or density violates positivity (Appendix A.3). Spatially heterogeneous disturbances and highly imbalanced variable distributions make accurate regression of

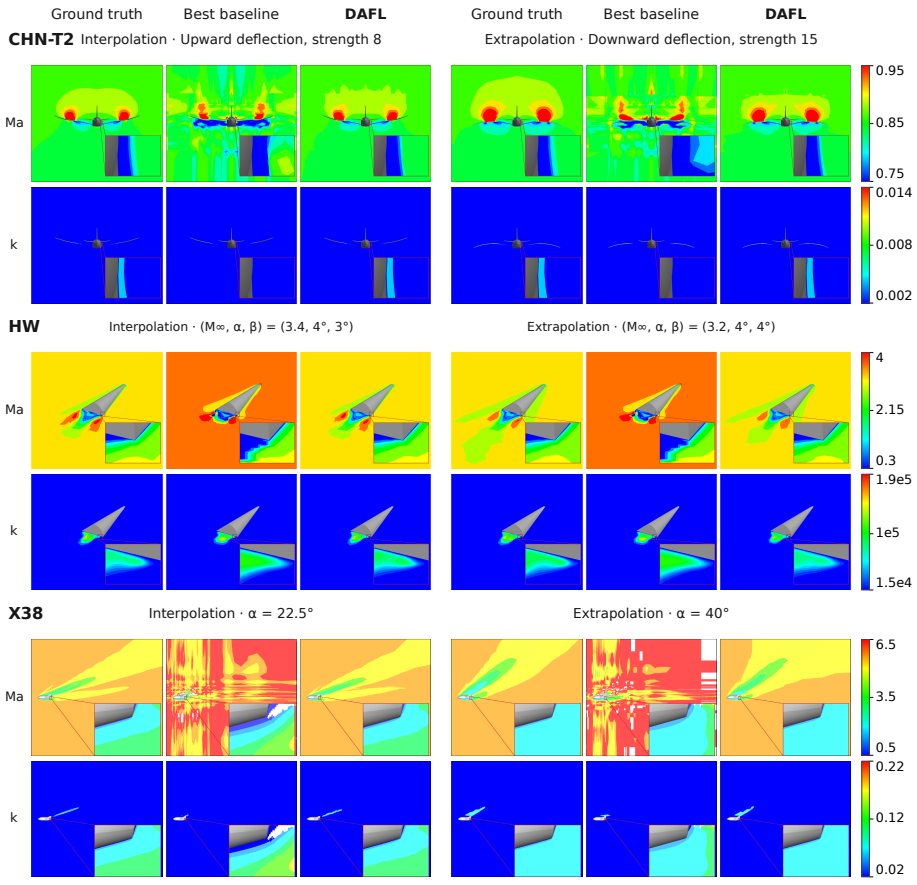


Figure 4: Comparison of Mach number and turbulent kinetic energy (k) from the ground truth, DAFL, and the best performing baseline for representative interpolation and extrapolation cases. Based on full field RL2 (Table 1), AB-UPT is selected for CHN-T2 and X38, and AdaField for HW. White regions indicate physically invalid predictions.

multiple flow variables difficult. DAFL reconstructs the Mach number and k fields more accurately than the best performing baselines across the interpolation and extrapolation cases. On CHN-T2 and X38, it reduces the banded Mach errors of AB-UPT and better preserves localized k structures. On X38, it also avoids the physically invalid regions indicated by white areas. On HW, DAFL more faithfully recovers the far-field Mach distribution and the extent of localized Mach and k structures than AdaField. The near-wall insets further demonstrate improved reconstruction of steep near-wall velocity gradients and k distributions.

4.3 CFD Warm-Starting and Solver Acceleration

For CFD warm-starting, the predicted initial states are evaluated for convergence robustness and solver acceleration. For each test case, iteration speedup is defined as

$$S = \frac{N_{\text{cold}}}{N_{\text{warm}}}. \tag{10}$$

Here, N_{cold} and N_{warm} denote the convergence iterations for cold-starting and warm-starting, respectively. Convergence is determined by the drag coefficient criterion in Appendix E. Speedup is computed only for cases achieving successful convergence under warm-starting, and the number of successfully converged cases is reported separately.

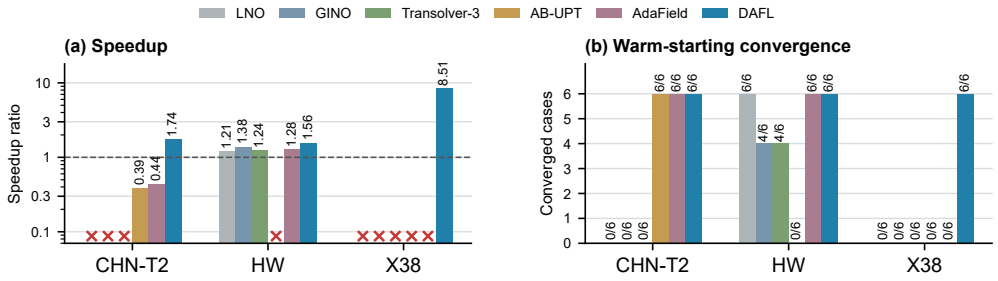


Figure 5: CFD warm-starting performance across the three datasets. (a) Solver speedup ratios, where crosses indicate cases that did not converge. (b) Number of converged cases out of six test cases.

DAFL achieved convergence under CFD warm-starting in all 18 test cases, including both interpolation and extrapolation cases (Figure 5). In every case, it required fewer iterations than cold-starting and any baseline that also converged. DAFL achieved mean iteration speedups of $1.74\times$, $1.56\times$, and $8.51\times$ on CHN-T2, HW, and X38, respectively.

By contrast, the baselines exhibited limitations in both convergence robustness and solver acceleration. For CHN-T2, only AB-UPT and AdaField converged under warm-starting, but both required more iterations than cold-starting in all six cases. For HW, GINO and Transolver-3 reduced the number of solver iterations in four of the six cases but failed to converge in the remaining two. LNO and AdaField converged in all six cases but required more iterations than DAFL, and LNO even required more iterations than cold-starting in one case (Appendix E.3). None of the five baselines converged for X38. Overall, these results demonstrate that DAFL enables robust CFD warm-starting and substantial solver acceleration across diverse geometries and flow conditions.

4.4 Ablation Study

We evaluate five ablations on X38, the most challenging dataset, to quantify the contribution of each DAFL component. X38 has the largest computational mesh, and none of the five baselines achieved convergence under CFD warm-starting. The full model achieved a full field RL2 of 0.125 (Figure 6). Each ablation increased RL2 relative to the full model in all six test cases (Appendix Table 13). Figure 7 compares Mach number fields computed from the predicted flow variables for the case at $\alpha = 22.5°$.

Voxel sampling. We replace DAFL's voxel sampling, performed separately in each predicted region, with random point sampling within those regions. This ablation increased the RL2 to 0.166, with larger errors along strong-disturbance boundaries and in the wake. These results indicate that the more uniform spatial coverage provided by voxel sampling improves the modeling of critical disturbances.

Disturbance-aware sampling and regression. We remove adaptive disturbance partitioning and use a single voxel size and regression head for the entire flow field. This ablation increased the RL2 to 2.415, the largest degradation among the five ablations. Errors in the extent and boundaries of the strong-disturbance region highlight the importance of adapting sampling and regression to disturbance characteristics.

Logarithmic wall distance. We remove the feature based on logarithmic wall distance from the model inputs. This ablation increased the RL2 to 0.437 and degraded the reconstruction of the near-wall Mach number layer. These results show that logarithmic wall distance improves the representation of steep near-wall velocity gradients.

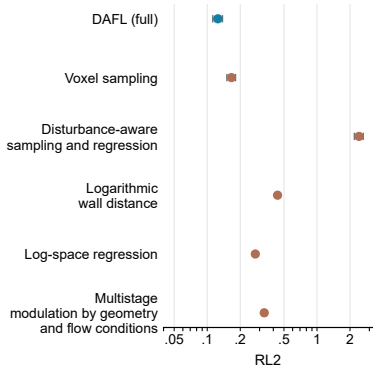


Figure 6: Ablation results on X38. Points indicate the full field RL2, with error bars showing the standard deviation across six test cases.

Log-space regression. We replace log-space regression and the exponential transformation with direct physical-space regression. This ablation increased the RL2 to 0.274 and produced structures with spuriously high Mach numbers and regions with physically invalid predictions. This result shows that log-space regression improves optimization for flow variables with diverse distributions and the exponential transformation preserves physically valid predictions.

Multistage modulation by geometry and flow conditions. We restrict modulation by geometry and flow conditions to the first encoder stage. This ablation increased the RL2 to 0.331 and produced errors in the spatial extent of the strong-disturbance region. This result highlights the importance of injecting geometry and flow conditions throughout the backbone to capture case-dependent variations.

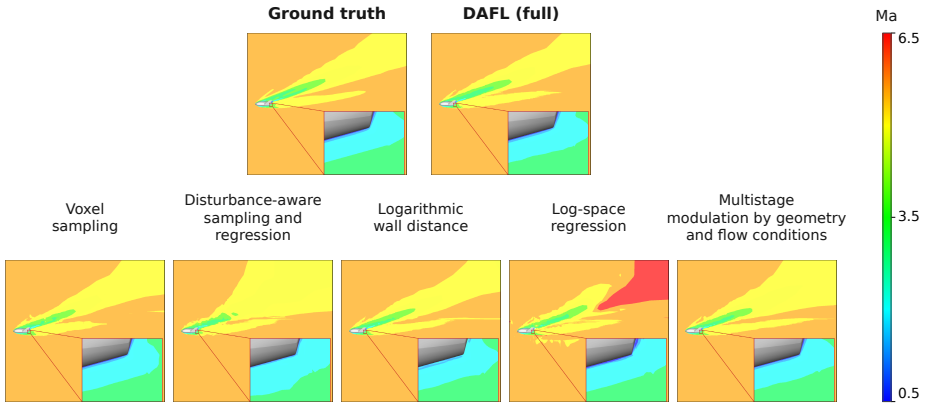


Figure 7: Mach number fields on X38 at $\alpha = 22.5°$ for the ground truth, full DAFL, and ablation variants. Insets highlight near-wall structures, and white regions indicate physically invalid predictions.

5 Conclusion

DAFL provides accurate and physically valid initial states for CFD warm-starting across diverse geometries and flow conditions by addressing spatially heterogeneous disturbances and highly imbalanced flow variable distributions. Across the complex 3D aircraft data, DAFL achieves the lowest full field RL2 and enables successful CFD warm-starting in all test cases, each requiring the fewest solver iterations.

Limitations and future work. The current adaptive disturbance partition relies on velocity and a preset threshold on wall distance. Future work could explore additional physical quantities to represent a broader range of disturbances. Conservation laws, boundary conditions, and constraints from the governing equations could also be incorporated to further improve physical consistency.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, manuscript formatting, figure formatting, and limited programming assistance, including code suggestions and debugging. We did not use generative AI tools to generate simulation data, conduct numerical experiments, select reported results, or interpret the experimental results. All text, code, and artifacts prepared with AI assistance were reviewed and verified by the authors. We take responsibility for the final content of this work, including content produced with the aid of generative AI.

REFERENCES

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.

Saakaar Bhatnagar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, and Shailendra Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64(2):525–545, 2019. doi: 10.1007/s00466-019-01740-0.

Florent Bonnet, Jocelyn Mazari, Paola Cinnella, and Patrick Gallinari. AirfRANS: High fidelity computational fluid dynamics dataset for approximating reynolds-averaged navier–stokes solutions. In *Advances in Neural Information Processing Systems: Datasets and Benchmarks Track*, 2022.

Xinhai Chen, Tiejun Li, Yunbo Wan, Yuechao Liang, Chunye Gong, Yufei Pang, and Jie Liu. Developing an advanced neural network and physics solver coupled framework for accelerating flow field simulations. *Engineering with Computers*, 40(2):1111–1126, 2024.

J. Gu, Y. Chen, Y. Bin, and S. Chen. GeoFormer: Mesh-free geometry-to-flow alignment framework for real-time aerodynamics on non-watertight vehicle geometries. *Physics of Fluids*, 38(1):015153, 2026.

Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 481–490, 2016. doi: 10.1145/2939672.2939738.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: A general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Thomas J. Horvath, Scott A. Berry, N. Ronald Merski, and Steve M. Fitzgerald. X-38 experimental aerothermodynamics. *Journal of Spacecraft and Rockets*, 41(2):272–292, 2004. doi: 10.2514/1.9198.

Jianing Huang, Kaixuan Zhang, Youjia Wu, and Ze Cheng. Operator learning with domain decomposition for geometry generalization in PDE solving. In *International Conference on Learning Representations*, 2026.

W. Hwang, S. Kim, D. Park, and S. Choi. A point cloud-based mesh-independent convolutional neural network framework for flow field prediction on variable geometries. *Physics of Fluids*, 37 (5):057126, 2025.

Roger L. Kimmel, David Adamczak, Allan Paull, Ross Paull, Jeremy Shannon, Robert Pietsch, Myles Frost, and Hans Alesi. HIFiRE-1 ascent-phase boundary-layer transition. *Journal of Spacecraft and Rockets*, 52(1):217–230, 2015. doi: 10.2514/1.A32851.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning-accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021. doi: 10.1073/pnas.2101784118.

J. Li, X. Zhang, W. Peng, X. Liu, W. Wang, and W. Yao. Accelerating aerodynamic simulations with a hybrid fine-tuned deep learning model. *Physics of Fluids*, 36(11):115181, 2024.

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

Zongyi Li, Nikola Kovachki, Christopher Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Max Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3d PDEs. In *Advances in Neural Information Processing Systems*, 2023b.

Hongyang Liu, Zhu Zhou, Yonggang Yu, Jiangtao Huang, Yu Tang, Chao Song, and Qingsheng Lan. Design of a standard aerodynamic model CHN-T2 for wide-body aircraft. *Acta Aerodynamica Sinica*, 41(10):15–29, 2023. doi: 10.7638/kqdlxxb-2023.0026.

Mark P. Loomis, Ethiraj Venkatapathy, Periklis Papadopoulos, Carol B. Davies, Scott Berry, Tom Horvath, and Charles Campbell. Aeroheating and aerodynamic CFD validation and prediction for the X-38 program. In *32nd Thermophysics Conference*, 1997. doi: 10.2514/6.1997-2478.

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

Sepehr Mousavi, Shizheng Wen, Levi Lingsch, Maximilian Herde, Bogdan Raonić, and Siddhartha Mishra. RIGNO: A graph-based framework for robust and accurate operator learning for PDEs on arbitrary domains. In *Advances in Neural Information Processing Systems*, 2025.

Octavi Obiols-Sales, Abhinav Vishnu, Nicholas Malaya, and Aparna Chandramowlishwaran. CFD-Net: A deep learning-based accelerator for fluid simulations. In *Proceedings of the 34th ACM International Conference on Supercomputing*, 2020.

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

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Louis Serrano, Lise Le Boudec, Armand Kassaï Koupaï, Thomas X. Wang, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. Operator learning with neural fields: Tackling PDEs on general geometries. In *Advances in Neural Information Processing Systems*, volume 36, pp. 70581–70611, 2023.

L. Stumpo, C. Alia, A. Ferrero, F. Masseni, et al. A graph neural network-based approach for initial solutions generation of hypersonic fields. In *27th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, 2026. doi: 10.2514/6.2026-5007.

Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58(1):25–36, 2020. doi: 10.2514/1.J058291.

Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6111–6122, 2020.

Tian Wang and Chuang Wang. Latent neural operator for solving forward and inverse PDE problems. In *Advances in Neural Information Processing Systems*, 2024.

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. In *Advances in Neural Information Processing Systems*, 2025.

Sebastian Willems, Ali Gülhan, Thomas J. Juliano, and Steven P. Schneider. Laminar to turbulent transition on the HIFiRE-1 cone at Mach 7 and high angle of attack. In *52nd Aerospace Sciences Meeting*, 2014. doi: 10.2514/6.2014-0428.

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*, 2024a.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4840–4851, 2024b.

Qisong Xiao, Xinhai Chen, Jie Liu, Chunye Gong, and Yufei Sun. Mh-dcnet: An improved flow field prediction framework coupling neural network with physics solver. *Computers & Fluids*, 284: 106440, 2024.

L. Xue, Y. Long, J. Wu, and T. Liu. A multi-scale, condition-adaptive point cloud framework for surface pressure prediction on complex aerodynamic geometries. *Physics of Fluids*, 38(1):017127, 2026.

X. Zhang, G. Sun, P. Zhang, Y. Wang, J. Zhang, L. Deng, et al. A residual graph convolutional network for setting initial flow field in computational fluid dynamics simulations. *Physics of Fluids*, 36(3):037150, 2024.

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

Junhong Zou, Wei Qiu, Zhenxu Sun, Xiaomei Zhang, Zhaoxiang Zhang, and Xiangyu Zhu. Adafield: Generalizable surface pressure modeling with physics-informed pre-training and flow-conditioned adaptation. *arXiv preprint arXiv:2601.07139*, 2026.

A Datasets and Evaluation Protocol

A.1 Benchmark Datasets and Splits

CHN-T2. CHN-T2 uses a wide-body aircraft geometry with a fuselage, supercritical wings, tail surfaces, and underwing nacelles. Its flow exhibits component interference, shock-induced separation, and boundary layer transition (Liu et al., 2023). We generate variants with upward and downward wing deflection, each with a corresponding structured volume mesh. The deflection magnitude is indexed by $s \in \{1, \ldots, 15\}$, with larger values indicating greater deflection. We hold $M_\infty = 0.85$, $\alpha = 3°$, and $\beta = 0°$ fixed while varying wing deflection to evaluate generalization across geometries.

HW. HW uses a hypersonic conical warhead geometry discretized with an unstructured volume mesh. Conical flows are common benchmarks for hypersonic aerothermodynamics and boundary layer transition (Kimmel et al., 2015). At nonzero angles of attack, the near-wall flow differs between the windward and leeward sides (Willems et al., 2014). We hold the geometry fixed across 32 training cases with different combinations of inflow Mach number M_∞, angle of attack α, and angle of sideslip β. This setup evaluates generalization across flow conditions.

X38. X38 uses the lifting-body geometry developed for NASA's X-38 Crew Return Vehicle program. At high angles of attack, the flow exhibits complex shock structures, viscous interactions, and pronounced near-wall variations (Loomis et al., 1997; Horvath et al., 2004). We use structured volume meshes for 19 training cases with different angles of attack α, while holding $M_\infty = 6$ and $\beta = 0°$ fixed. This setup evaluates generalization across angles of attack.

Each dataset contains six test cases. Interpolation cases have parameter values within the training ranges, whereas extrapolation cases have at least one parameter value outside those ranges. Table 2 summarizes the training sets and all 18 test cases.

Table 2: Training sets and test cases. The final parenthesized number in each entry gives the case count. For CHN-T2, each value of s corresponds to one upward and one downward wing deflection. The HW training set contains all combinations of the listed values. HW test tuples follow the order $(M_\infty, \alpha, \beta)$.

Dataset	Training set	Test set	
		Interpolation	Extrapolation
CHN-T2	$s \in \{2, \ldots, 7, 9, \ldots, 14\}$ (24)	$s = 8$ (2)	$s \in \{1, 15\}$ (4)
HW	$M_\infty \in \{3.2, 3.6, 4.0, 4.4\}$, $\alpha \in \{2°, 4°, 6°, 8°\}$, $\beta \in \{1°, 3°\}$ (32)	$(3.4, 4°, 3°)$ $(3.6, 3°, 1°)$ $(3.2, 4°, 2°)$ (3)	$(4.8, 4°, 3°)$ $(4.0, 0°, 1°)$ $(3.2, 4°, 4°)$ (3)
X38	$\alpha \in \{21°, 22°, \ldots, 39°\}$ (19)	$\alpha = 22.5°, 28.5°, 34.5°, 36.5°$ (4)	$\alpha \in \{20°, 40°\}$ (2)

A.2 Output Variables and Metrics

A.2.1 Disturbance Partition Metrics

For each class $c \in \{\text{weak}, \text{strong}\}$, let TP_c, FP_c, and FN_c denote the numbers of true positives, false positives, and false negatives, respectively. We quantify the spatial overlap between the predicted and reference regions for each class using intersection over union (IoU):

$$\text{IoU}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c}, \qquad \text{mIoU} = \frac{1}{2} \sum_c \text{IoU}_c. \tag{11}$$

Mean IoU (mIoU) is the average of the IoU values for the two classes. It gives equal weight to the weak- and strong-disturbance classes. This reduces the effect of class imbalance compared with overall accuracy.

We use recall to measure the fraction of reference points correctly identified in each class:

$$R_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}. \tag{12}$$

For the strong-disturbance class, false negatives are reference strong-disturbance points assigned to the weak-disturbance region.

We also report the F1 score, the harmonic mean of precision and recall:

$$F1_c = \frac{2\mathrm{TP}_c}{2\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c}. \tag{13}$$

The F1 score accounts for both missed points and false positives. All metrics are dimensionless. Higher IoU, mIoU, and F1 scores indicate better agreement between predicted and reference partitions. Higher recall indicates that a larger fraction of reference points is recovered.

A.2.2 Full Field Prediction Metrics

For CHN-T2 and X38, we predict eight variables: $\{\rho, u, v, w, p, k, \omega, \mu_t\}$. For HW, we also predict temperature T. We use relative L_2 error (RL2) as the primary metric for full field prediction. We also report mean squared error (MSE) and mean absolute error (MAE) for each variable.

Relative L_2 error. For each output variable, let $\mathbf{y}, \widehat{\mathbf{y}} \in \mathbb{R}^N$ denote the ground-truth and predicted fields at N cell centers, respectively. RL2 is defined as

$$\mathrm{RL2}(\mathbf{y}, \widehat{\mathbf{y}}) = \frac{\|\widehat{\mathbf{y}} - \mathbf{y}\|_2}{\|\mathbf{y}\|_2}. \tag{14}$$

RL2 is dimensionless, and lower values indicate more accurate predictions. We obtain the full field RL2 for each dataset by averaging across output variables within each case and then across the six test cases.

Mean squared and absolute errors. For each output variable, MSE and MAE are defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i - y_i)^2, \qquad \mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\widehat{y}_i - y_i|. \tag{15}$$

MSE is more sensitive to large errors than MAE, and both metrics depend on the variable's units and scale. For each variable, we report MSE and MAE averaged over the six test cases.

A.3 Visualization Protocol

We visualize Mach number and turbulent kinetic energy k to compare predicted flow structures. Mach number combines predictions of velocity, pressure, and density, while k highlights the intensity and spatial distribution of turbulence.

Local Mach number. At each cell center, we calculate the local Mach number from the velocity components (u, v, w), static pressure p, and density ρ:

$$\mathrm{Ma} = \frac{\sqrt{u^2 + v^2 + w^2}}{\sqrt{\gamma p/\rho}}, \qquad \gamma = 1.4. \tag{16}$$

Here, γ is the ratio of specific heats for the gas model. We compute Mach number from the ground-truth field and each predicted field separately. White regions indicate invalid predictions for which Mach number cannot be computed.

Turbulent kinetic energy. We visualize k directly from the ground-truth and predicted fields.

B Implementation and Training Details

B.1 Binary Disturbance Label Generation

We use the same rule to generate binary disturbance labels for all three datasets. For each case, $\mathbf{v} = (u, v, w)$ denotes the ground-truth velocity at each point. The inflow velocity is $\mathbf{v}_\infty = (u_\infty, v_\infty, w_\infty)$. We calculate the velocity disturbance $\Delta\mathbf{v} = \mathbf{v} - \mathbf{v}_\infty$ as in Eq. equation 1.

For each velocity component of $\Delta\mathbf{v}$, we fit a separate Gaussian mixture model (GMM) with three mixture components. Each point is assigned to its most probable component in each GMM. For each velocity component, points assigned to the GMM component with the largest mean form the strong-disturbance mask. We combine the three masks using a logical OR:

$$\mathbf{b}^\star = \mathbf{m}_u \vee \mathbf{m}_v \vee \mathbf{m}_w. \tag{17}$$

Points in at least one mask receive a strong-disturbance label ($b_i^\star = 1$). All other points receive a weak-disturbance label ($b_i^\star = 0$).

Algorithm 1 Binary disturbance label generation using GMMs

Require: Ground-truth velocity field $\mathbf{v} = (u, v, w)$ and inflow velocity $\mathbf{v}_\infty = (u_\infty, v_\infty, w_\infty)$ for one case
Ensure: Binary disturbance labels $\mathbf{b}^\star$
1: $\Delta\mathbf{v} \leftarrow \mathbf{v} - \mathbf{v}_\infty$
2: $\mathbf{b}^\star \leftarrow \mathbf{0}$
3: For each velocity component of $\Delta\mathbf{v}$, fit a separate GMM with three mixture components
4: Assign each point to its most probable component in each GMM
5: $\mathbf{m}_u, \mathbf{m}_v, \mathbf{m}_w \leftarrow$ masks of points assigned to the component with the largest mean in each GMM
6: $\mathbf{b}^\star \leftarrow \mathbf{m}_u \vee \mathbf{m}_v \vee \mathbf{m}_w$
7: **return** $\mathbf{b}^\star$

B.2 Three-Region Construction and Sampling Scales

The binary disturbance partition first separates each flow field into weak- and strong-disturbance regions. Strong-disturbance points with $d_i < \tau$ form the near region. The remaining strong-disturbance points, with $d_i \geq \tau$, form the outer region (Eq. equation 3).

Table 3 reports the threshold τ on wall distance and the voxel sizes for each dataset. Each region is voxelized separately using its corresponding size. In all three datasets, $h_{\text{near}} < h_{\text{outer}} < h_{\text{weak}}$. This ordering gives the near region the finest sampling resolution and the weak-disturbance region the coarsest.

Table 3: Thresholds on wall distance and voxel sizes for each region (mm).

Dataset	τ	h_{weak}	h_{near}	h_{outer}
CHN-T2	350	300	3	100
HW	4	4	0.2	2
X38	11	2	0.2	1

B.3 Network Architecture and Training Settings

Network design. DAFL uses a disturbance partition predictor and a full field regressor. Both networks use Point Transformer V3 (PTv3) (Wu et al., 2024b) backbones with the same point inputs and modulation by geometry and flow conditions. The two networks have separate parameters. Table 4 summarizes their shared components and the settings for each task.

Table 4: Shared components and task specific settings of the two DAFL networks.

Component	Disturbance partition predictor	Full field regressor
Task	Binary classification of weak and strong disturbances	Full field multivariable regression
Point input	Coordinates $\mathbf{x}_i$ and logarithmic wall distance $\phi(d_i)$	Coordinates $\mathbf{x}_i$ and logarithmic wall distance $\phi(d_i)$
Global condition	Surface geometry $\mathcal{S}$ and flow condition $\mathbf{c}$	Surface geometry $\mathcal{S}$ and flow condition $\mathbf{c}$
Backbone	PTv3	PTv3
Conditioning	Modulation by geometry and flow conditions in encoder and decoder stages	Modulation by geometry and flow conditions in encoder and decoder stages
Head and output	Linear classifier producing $\mathbf{p}_i$ and $\widehat{b}_i$	Three independent $128 \to 128 \to Q$ heads with GELU ($Q = 8$ for CHN-T2 and X38, $Q = 9$ for HW)
Target and loss	Labels $b_i^\star$ with cross-entropy and Dice losses	Positive flow variables in log space and velocity components in physical space, with regional mean squared error

PTv3 settings. The two networks use identical PTv3 configurations, as detailed in Table 5.

Table 5: PTv3 backbone settings shared by both networks.

Component	Configuration
Version	PT-v3m2
Encoder	Five stages; one block per stage; 256 channels; attention heads $(2, 4, 8, 16, 32)$
Decoder	Four stages; one block per stage; 128 channels; attention heads $(2, 4, 8, 16)$
Spatial hierarchy	Downsampling strides $(2, 2, 2, 2)$; Z-order serialization; patch size 1024
Transformer block	MLP ratio 4; pre-normalization; Q/K normalization; FlashAttention
Regularization	Maximum drop-path rate of 0.3

Targets and training. The full field regressor predicts ρ, p, k, ω, and μ_t in log space. For HW, it also predicts temperature T in log space. An exponential transformation maps these predictions back to physical space and ensures positive values for CFD warm-starting (Eq. equation 9). The velocity components are predicted in physical space. Both networks and all ablation models use the training settings in Table 6.

Table 6: Training settings for the DAFL networks and ablation models.

Setting	Value
Optimizer	AdamW with $\beta = (0.9, 0.95)$ and weight decay 0.05
Peak learning rate	2×10^{-4}
Schedule	OneCycleLR with cosine annealing and a 10% warm-up
Epochs	500
Batch size	8
Gradient clipping	Global norm capped at 1.0
Hardware	One NVIDIA A100 GPU (40 GB)

Condition encoding. The representation $\mathbf{g}$ combines the feature $\mathbf{g}_\mathrm{s}$ encoding surface geometry and the feature $\mathbf{g}_\mathrm{c}$ encoding flow conditions (Eq. equation 6).

Let $\mathcal{S} = \{\mathbf{s}_j\}_{j=1}^{N_\mathrm{s}}$ denote the surface point cloud. We apply sinusoidal positional encoding to each coordinate of every surface point:

$$\begin{aligned} \gamma(t) &= \big[\sin(2^\ell \pi t)\big]_{\ell=0}^{K-1} \,\|\, \big[\cos(2^\ell \pi t)\big]_{\ell=0}^{K-1}, \\ \mathrm{PE}(\mathbf{s}_j) &= \gamma(s_{j,x}) \,\|\, \gamma(s_{j,y}) \,\|\, \gamma(s_{j,z}). \end{aligned} \tag{18}$$

Here, $s_{j,x}$, $s_{j,y}$, and $s_{j,z}$ are the coordinates of $\mathbf{s}_j$. With $K = 4$, the angular frequencies are π, 2π, 4π, and 8π. The symbol $\|$ denotes feature concatenation.

An MLP maps each positional encoding to a point feature. We average features within each surface voxel before applying a sparse convolutional network. Global average and max pooling summarize the resulting features. A geometry MLP maps the pooled features to $\mathbf{g}_\mathrm{s}$. Another MLP maps $\mathbf{c} = (M_\infty, \alpha, \beta)$ to $\mathbf{g}_\mathrm{c}$. We concatenate these two features to obtain $\mathbf{g}$:

$$\begin{aligned} \mathbf{H}_\mathrm{s} &= \mathrm{SConv}_{\theta_\mathrm{s}}\left(\mathrm{VoxelAvg}_{h_\mathrm{s}}\left(\{\mathrm{MLP}_\mathrm{pos}\left(\mathrm{PE}(\mathbf{s}_j)\right)\}_{j=1}^{N_\mathrm{s}}\right)\right), \\ \mathbf{g}_\mathrm{s} &= E_\mathrm{s}(\mathcal{S}) = \mathrm{MLP}_\mathrm{geo}\left(\mathrm{AvgPool}(\mathbf{H}_\mathrm{s}) \parallel \mathrm{MaxPool}(\mathbf{H}_\mathrm{s})\right), \\ \mathbf{g}_\mathrm{c} &= \mathrm{MLP}_\mathrm{cond}\left([M_\infty, \alpha, \beta]^\mathsf{T}\right), \qquad \mathbf{g} = \mathbf{g}_\mathrm{s} \parallel \mathbf{g}_\mathrm{c}. \end{aligned} \tag{19}$$

Here, $\mathrm{VoxelAvg}_{h_\mathrm{s}}$ averages features within surface voxels of size h_s. The operator $\mathrm{SConv}_{\theta_\mathrm{s}}$ denotes the sparse convolutional network. Its output $\mathbf{H}_\mathrm{s}$ is a local surface feature map. All points in the volume mesh of a case share $\mathbf{g}$. Equation equation 7 injects geometry and flow conditions into every stage of both backbones.

B.4 Baseline Adaptation

We adapt all five baselines to predict full flow fields at CFD cell centers. Within each dataset, they predict the same variables and are evaluated on the same test cases. The methods include neural operators and Transformers that operate on point clouds.

LNO. Latent Neural Operator (LNO) (Wang & Wang, 2024) uses Physics-Cross-Attention to map irregular input points to a fixed set of latent modes. Attention blocks process these modes, and inverse cross-attention decodes values at query points. CFD cell centers serve as both input and query points. Wall distance and the flow condition are included as input features. The output layer predicts the full flow field. We use a width of 512, eight attention blocks, eight attention heads, and 64 latent modes.

GINO. Geometry-Informed Neural Operator (GINO) (Li et al., 2023b) combines a Graph Neural Operator (GNO) with a Fourier Neural Operator (FNO). The input GNO maps irregular points to a regular latent grid. The FNO processes the grid features, and the output GNO maps them to query points. CFD cell centers serve as both input and query points. Wall distance and the flow condition are included as input features. The latent grid contains 8^3 points. The FNO has eight layers, a width of 256, and four Fourier modes along each axis. Both GNOs use a neighborhood radius of 0.35.

Transolver-3. Transolver-3 (Zhou et al., 2026) uses learned slice weights to group mesh points into a small set of physical states. Self-attention processes these states before mapping them back to the mesh points. Geometry slice tiling computes slice weights in blocks to reduce memory use. The input features are coordinates, wall distance, and the flow condition. The output dimension matches the number of predicted variables in each dataset. We use a width of 512, 16 layers, eight attention heads, 64 slices, and a block size of 4,096.

AB-UPT. Anchored-Branched Universal Physics Transformers (AB-UPT) (Alkin et al., 2025) use separate branches for geometry, surface fields, and volume fields. The geometry branch maps the surface point cloud to geometry supernodes. The two branches for physical fields use anchors and exchange information through cross-attention. We provide the surface point cloud and flow condition as inputs. Cell centers in the volume mesh serve as query points, and the volume branch predicts the full flow field. We use a width of 256, four attention heads, eight blocks per physical branch, at most 4,096 anchors, and 512 geometry supernodes.

AdaField. AdaField (Zou et al., 2026) uses a Semantic Aggregation Point Transformer (SAPT) with local vector attention and semantic aggregation. Its Flow-Conditioned Adapter incorporates the flow condition. For our full field task, we retain the surface encoder and condition adapter and use a cross-attention decoder for queries on the volume mesh. The surface point cloud represents geometry. Cell centers in the volume mesh and their wall distances serve as query inputs. The encoder channels are $(64, 128, 256, 512)$, with four blocks per stage. The decoder has four blocks and eight attention heads. We use at most 4,096 surface points as input.

C SUPPLEMENTARY EVIDENCE FOR DISTURBANCE-AWARE FIELD LEARNING

This section presents supplementary analyses of disturbance partitioning and representation of flow variables. We first compare GMM-derived and predicted disturbance partitions across geometries and flow conditions. We then examine regional distributions of flow variables and the logarithmic representation of wall distance. Finally, we examine log-space regression and the positivity of the transformed predictions.

C.1 ADAPTIVE DISTURBANCE PARTITIONS ACROSS GEOMETRY AND FLOW CONDITIONS

Figure 8 compares ground-truth Mach number fields, GMM-derived reference partitions, and predicted partitions for two cases from each dataset. CHN-T2 varies wing deflection under a fixed flow condition. HW and X38 vary flow conditions with fixed geometries.

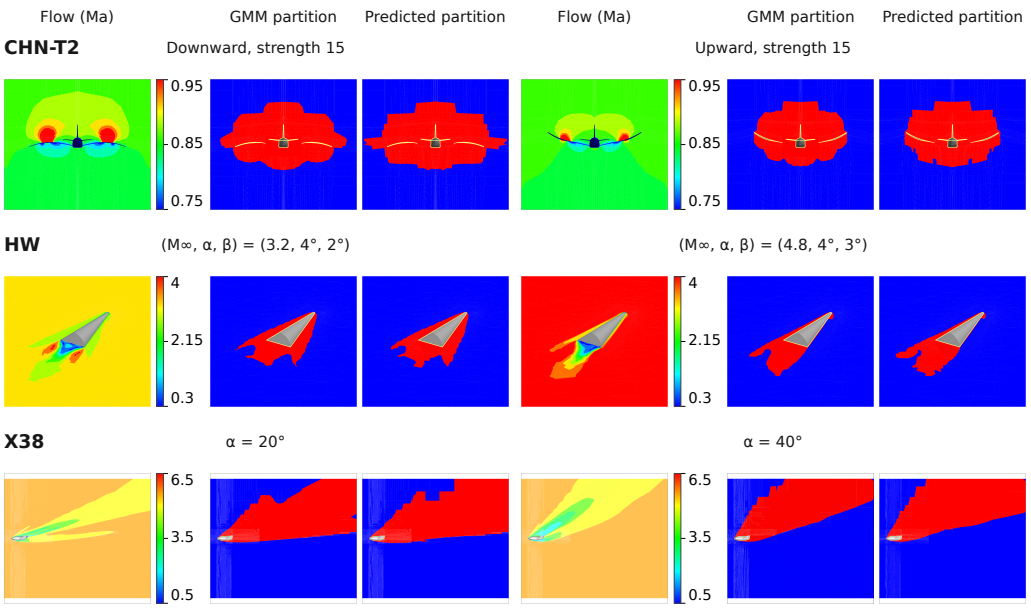


Figure 8: Ground-truth Mach number fields, GMM-derived reference partitions, and predicted partitions for two cases from each dataset. Red and blue denote strong and weak disturbances, respectively. For CHN-T2, the left and right cases have downward and upward wing deflections at strength 15. For HW, the left and right cases are $(M_\infty, \alpha, \beta) = (3.2, 4^\circ, 2^\circ)$ and $(4.8, 4^\circ, 3^\circ)$. For X38, they are $\alpha = 20^\circ$ and $\alpha = 40^\circ$.

In the cases shown, the GMM-derived strong-disturbance regions occur mainly where the Mach number differs from the inflow Mach number. Regions far from the body are mostly labeled as weak disturbances. The predicted partitions closely match the GMM-derived reference partitions.

In CHN-T2, reversing wing deflection changes the strong-disturbance region around both wings. In HW, varying M_∞, α, and β changes its extent around and downstream of the cone. In X38, increasing α from 20° to 40° shifts the strong-disturbance region upward. These case-dependent changes support conditioning the partition predictor on geometry and flow conditions.

C.2 DISTURBANCE PARTITION PREDICTION

Table 7 compares the predicted partitions with the GMM-derived reference labels using mIoU, IoU, recall, and F1.

Table 7: Disturbance partition prediction results. Higher values indicate better performance.

		Weak disturbance			Strong disturbance		
Dataset	mIoU	IoU	Recall	F1	IoU	Recall	F1
CHN-T2	0.876	0.838	0.931	0.912	0.913	0.944	0.955
HW	0.956	0.935	0.951	0.966	0.978	0.994	0.989
X38	0.979	0.976	0.984	0.988	0.982	0.994	0.991

The predictor achieves mIoU values of 0.876, 0.956, and 0.979 on CHN-T2, HW, and X38, respectively. CHN-T2 has the lowest mIoU. Its varying geometry may make partition prediction more difficult. Strong-disturbance recall is 0.944 on CHN-T2 and 0.994 on both HW and X38. These values show that the predictor identifies most reference strong-disturbance points in every dataset. The representative examples in Figure 8 show similar spatial patterns between predicted and GMM-derived partitions.

C.3 Flow variable distributions across three regions

Figure 9 compares flow variable distributions in the weak, near, and outer regions for one case from each dataset. The GMM-derived label $b_i^\star$ identifies weak ($b_i^\star = 0$) and strong ($b_i^\star = 1$) disturbances. Strong-disturbance points are then divided into near and outer regions using the criterion based on wall distance in Eq. equation 3. The thresholds for each dataset are listed in Table 3.

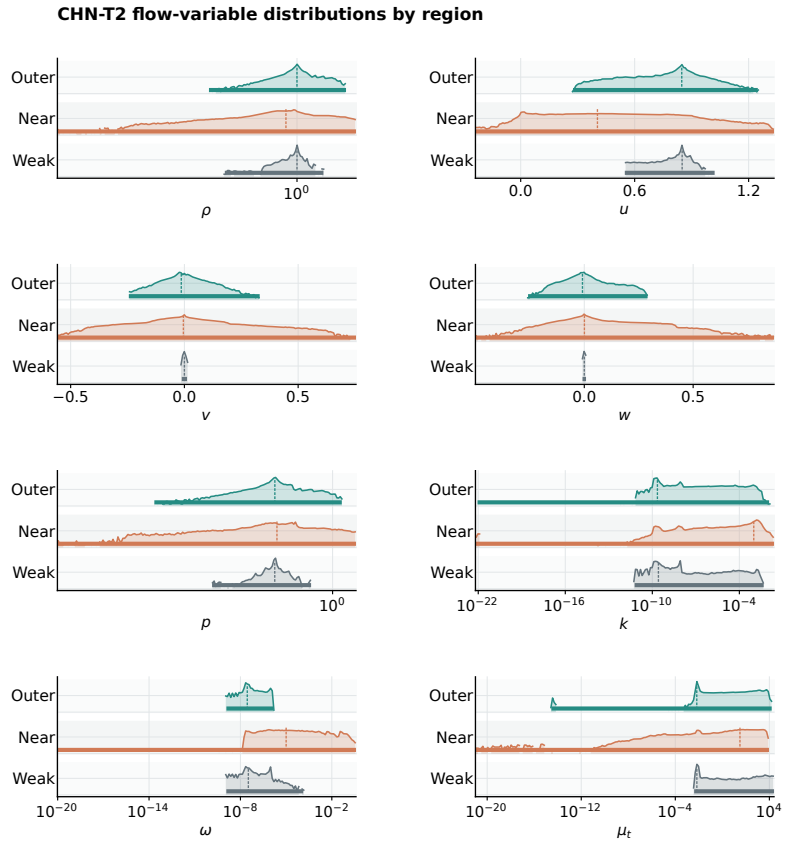


Figure 9: Flow variable distributions in the weak, near, and outer regions for a CHN-T2 case with downward deflection at strength 8. The threshold on wall distance is $\tau = 350\,\mathrm{mm}$. Within each panel, the three regions are shown from bottom to top. The horizontal axis shows variable values. For each region, the height of the curve shows the percentage of its points in each bin. Solid horizontal lines mark the observed ranges, and dashed vertical lines mark the medians.

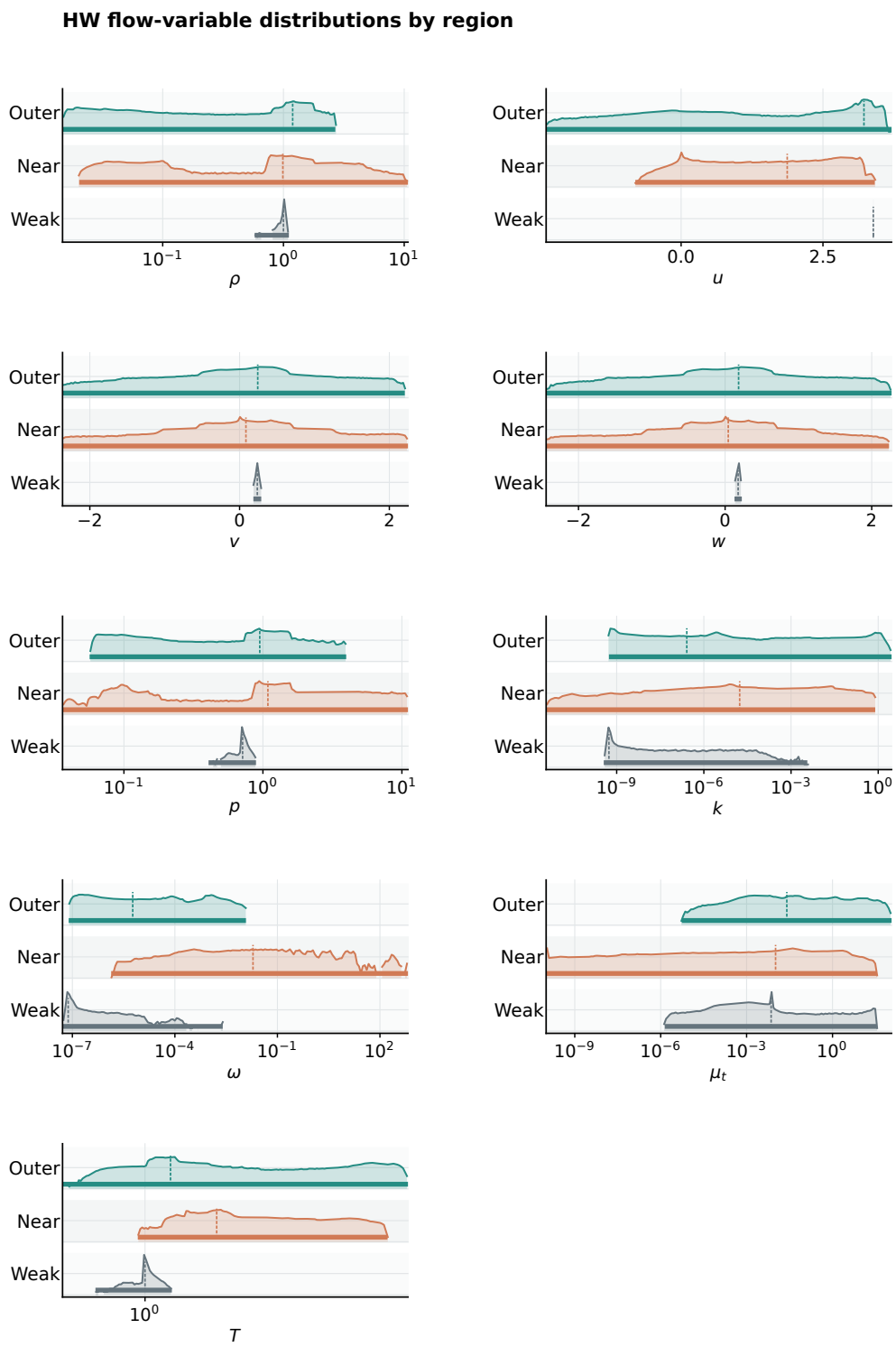


Figure 9: Continued. Flow variable distributions for an HW case at $(M_\infty, \alpha, \beta) = (3.4, 4^\circ, 3^\circ)$, with $\tau = 4\,\mathrm{mm}$. Temperature T is shown in the last row.

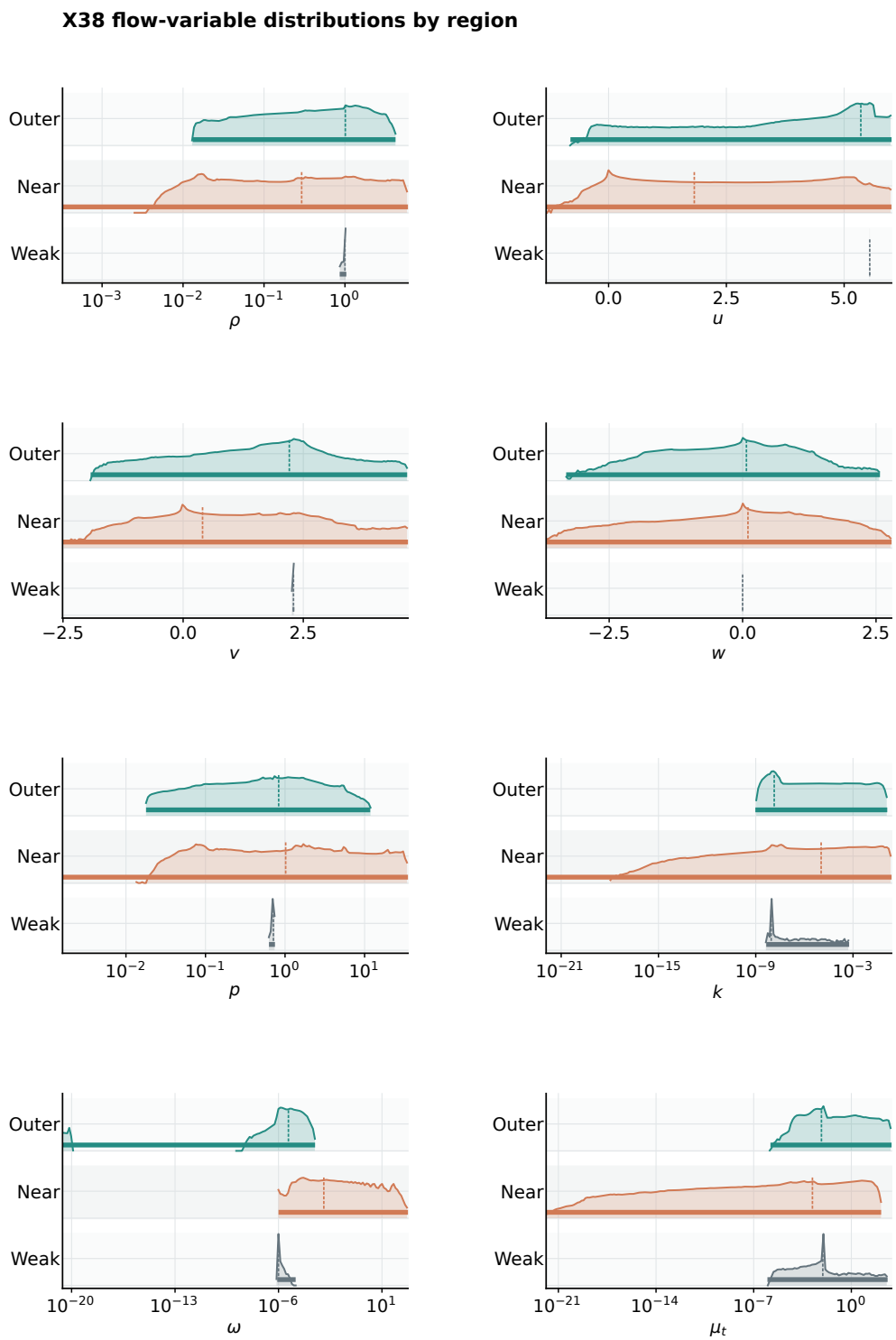


Figure 9: Continued. Flow variable distributions for an X38 case at $M_\infty = 6$ and $\alpha = 22.5°$, with $\tau = 11\,\mathrm{mm}$.

Across the three cases, density, pressure, and velocity components in the weak region are concentrated near their inflow values. Their distributions in the near and outer regions are generally broader. Temperature in HW shows a similar pattern. In all three cases, the near region has a lower median u and higher median k and ω than the outer region. Together, these differences motivate coarser sampling in the weak region, finer sampling in the strong-disturbance regions, and separate regression heads for the three regions.

C.4 Logarithmic representation of wall distance

Figure 10 compares the distributions of wall distance d_i and its logarithmic representation $\ln(d_i)$ for one case from each dataset.

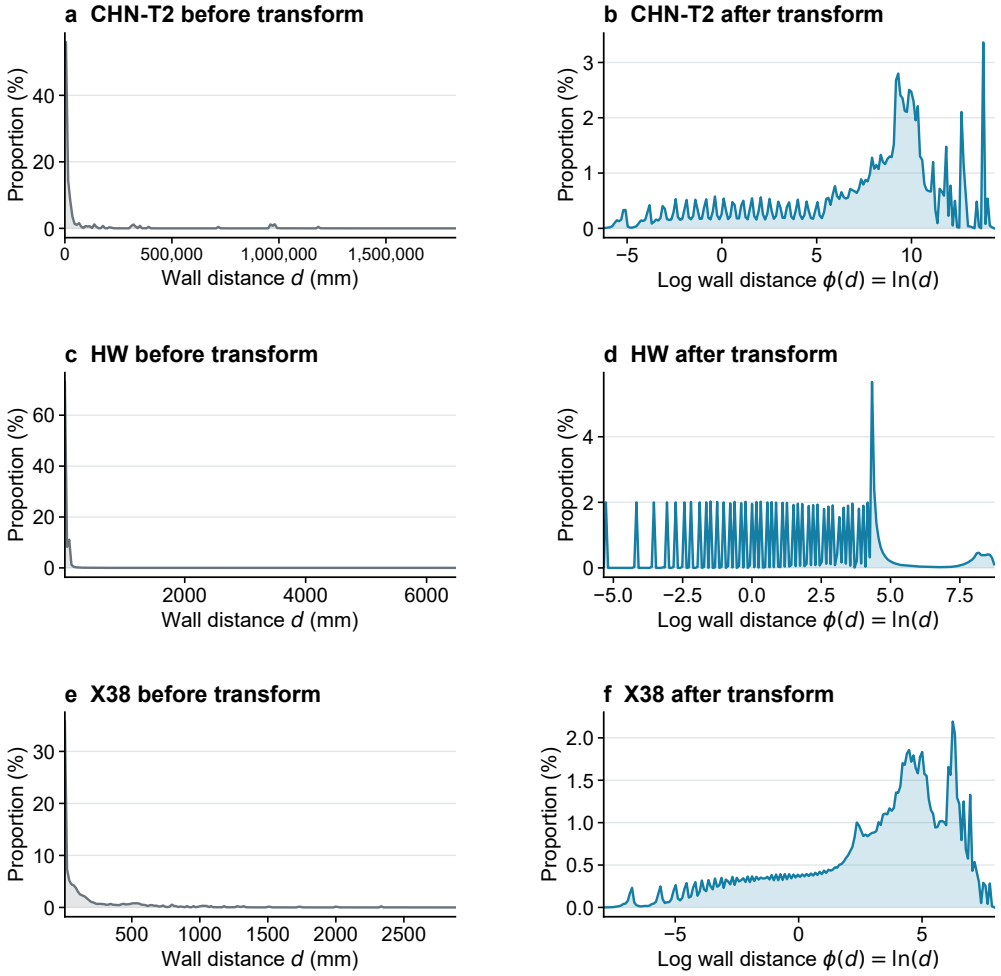


Figure 10: Distributions of raw wall distance d_i (left) and logarithmic wall distance $\ln(d_i)$ (right) for one case each from CHN-T2 (a,b), HW (c,d), and X38 (e,f). The vertical axis shows the percentage of points in each bin. The three cases contain 1,091,432, 1,645,295, and 7,412,224 points, respectively.

In all three cases, the transformation spreads small wall distances over a wider range of feature values and compresses large wall distances. This wider separation helps the model represent steep near-wall velocity gradients.

C.5 IMBALANCED FLOW VARIABLE DISTRIBUTIONS AND LOG-SPACE REGRESSION

Figure 11 compares the distributions of physical values and corresponding training targets for one case from each dataset. The transformation $z = \ln y$ is applied to ρ, p, k, ω, and μ_t, and to T in HW. Velocity components are used as targets without transformation.

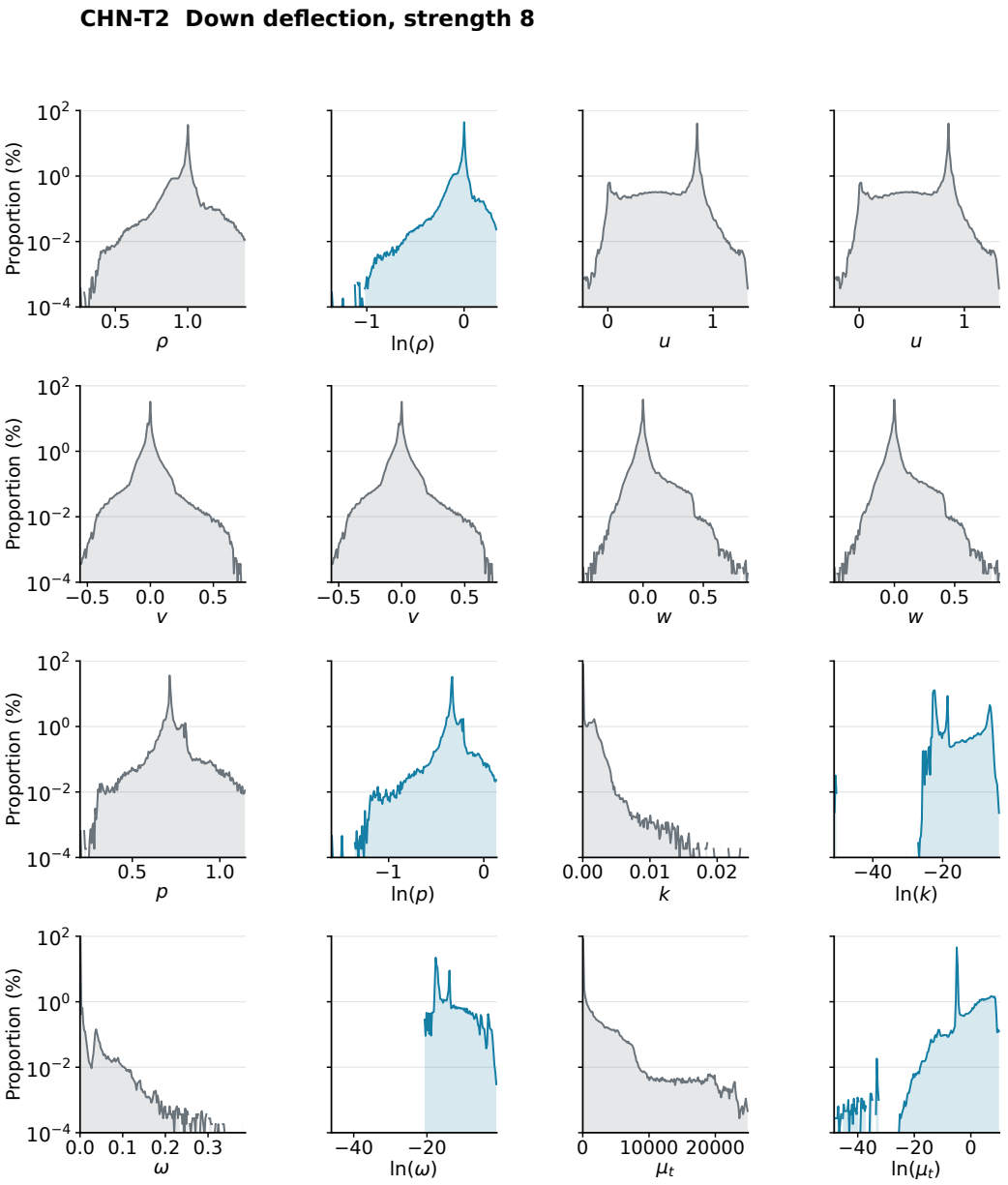


Figure 11: Distributions of physical values and training targets for a CHN-T2 case with downward deflection at strength 8. Each pair shows physical values on the left and the corresponding targets on the right. The transformation $z = \ln y$ is applied to ρ, p, k, ω, and μ_t. Velocity components are unchanged. The horizontal axes cover the full observed range of each quantity. The vertical axes show the percentage of points in each bin.

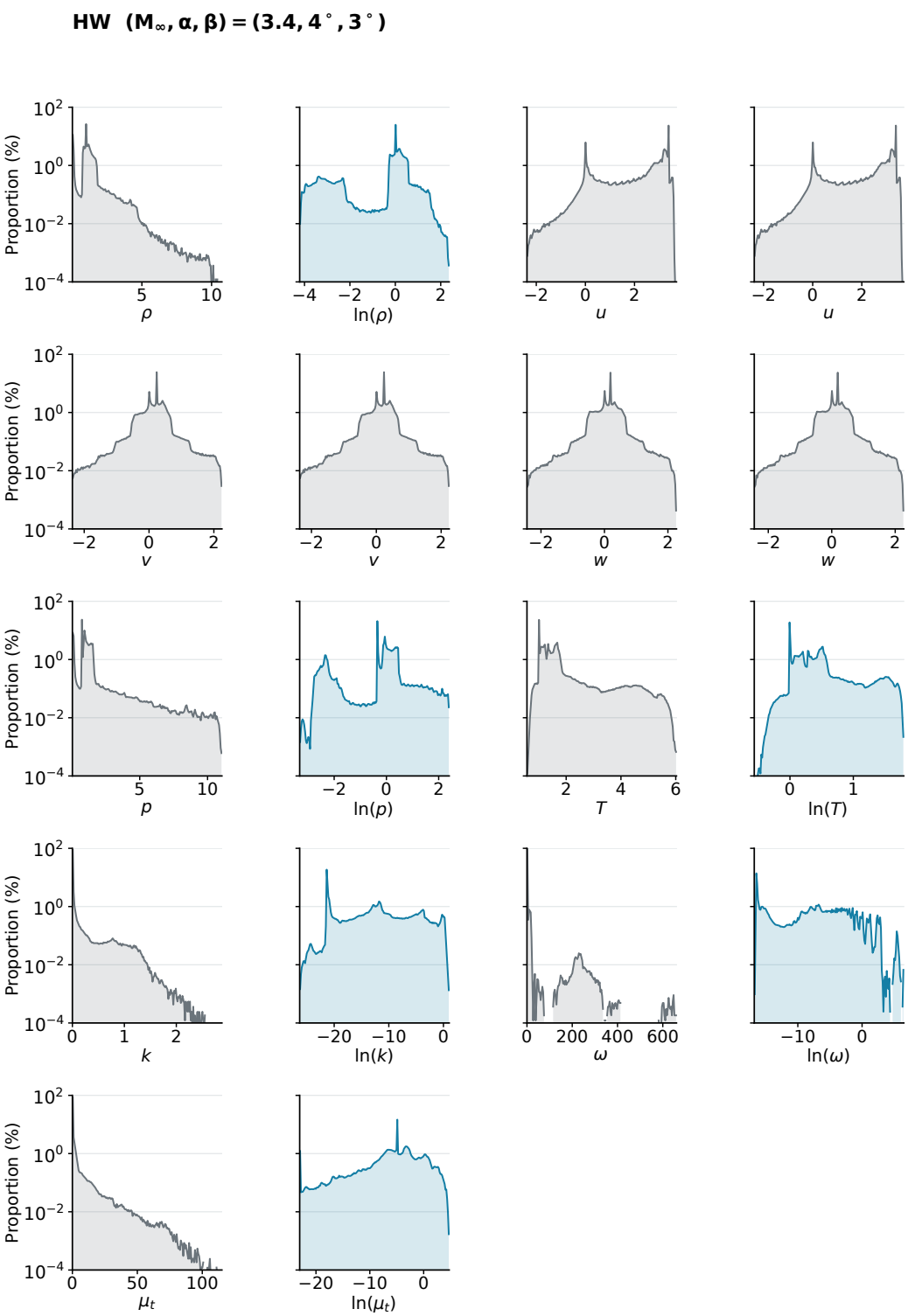


Figure 11: Continued. Distributions of physical values and training targets for an HW case at $(M_\infty, \alpha, \beta) = (3.4, 4^\circ, 3^\circ)$. Temperature T is also transformed as $z = \ln y$.

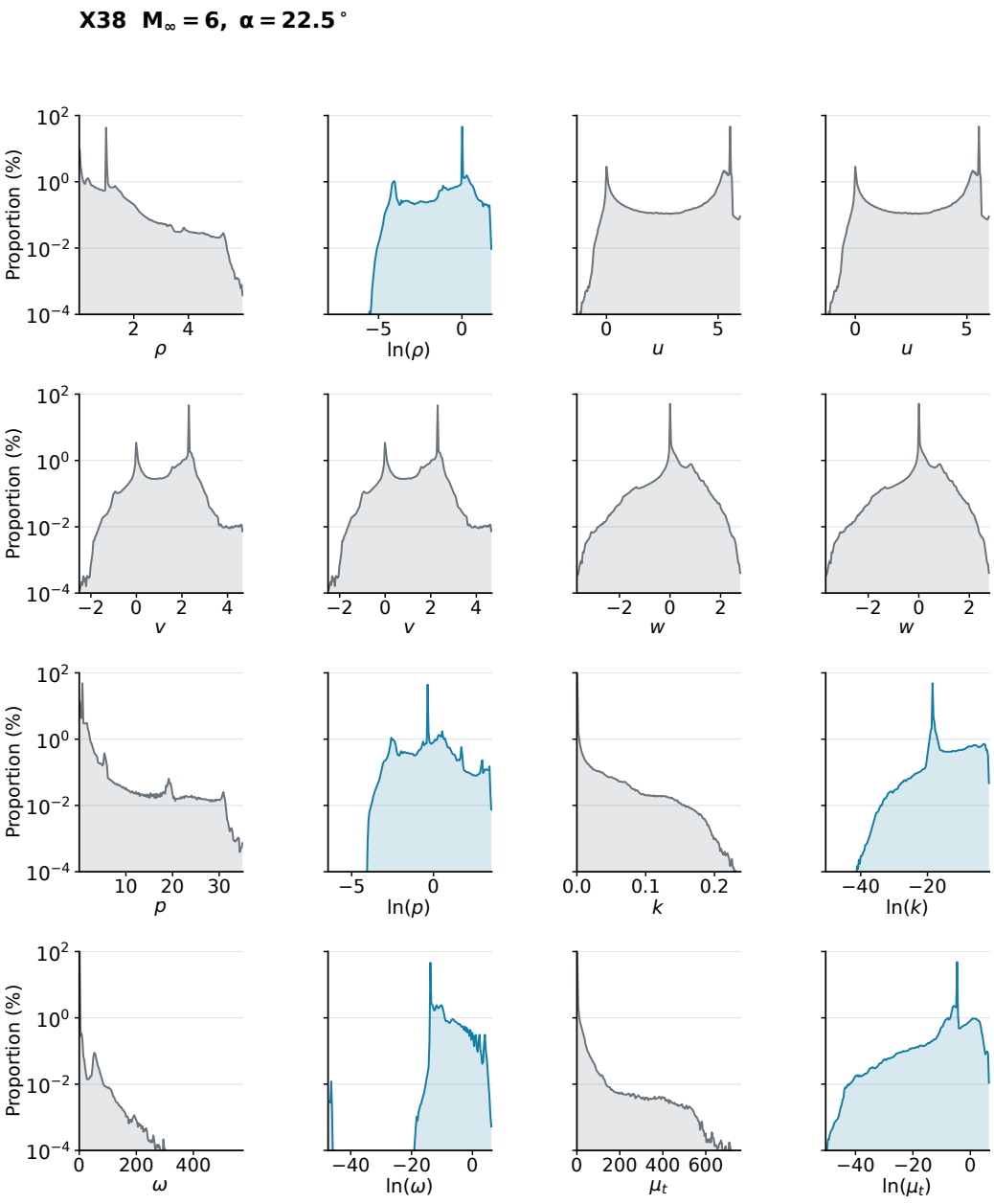


Figure 11: Continued. Distributions of physical values and training targets for an X38 case at $\alpha = 22.5°$.

For the cases shown, the observed ranges of ρ, p, and HW temperature T span 0.74–4.34 orders of magnitude. The ranges of k, ω, and μ_t span 10.1–25.4 orders of magnitude. Low values of the turbulence variables are common, whereas high values are rare. The logarithmic transformation spreads low values over a wider target range and compresses high values. This can ease optimization for variables with wide value ranges. Velocity components are regressed in physical space because their distributions are less imbalanced and their values can be positive or negative. The exponential transformation maps log-space predictions back to physical space and guarantees positive outputs for the transformed variables. This positivity is necessary for physically valid initial states in CFD warm-starting.

D FULL FIELD PREDICTION RESULTS

DAFL achieves the lowest full field RL2 on all three datasets (Table 1). This section analyzes performance across test cases and physical variables, then compares the spatial structures of the predicted fields.

D.1 ACCURACY ACROSS TEST CASES

Table 8: Full field RL2 (↓) for all 18 test cases. I and E denote interpolation and extrapolation. CHN-T2 cases are identified by deflection direction and strength. HW cases are identified by $(M_\infty, \alpha, \beta)$. X38 cases are identified by the angle of attack α.

Condition	Setting	LNO	GINO	Transolver-3	AB-UPT	AdaField	DAFL
CHN-T2							
Downward, strength 8	I	1.74	1.03	2.94	0.496	0.536	0.185
Upward, strength 8	I	2.25	1.06	3.24	0.475	0.517	0.384
Downward, strength 1	E	2.21	1.06	3.06	0.484	0.520	0.208
Downward, strength 15	E	1.28	1.01	2.81	0.523	0.550	0.213
Upward, strength 1	E	2.23	1.06	3.12	0.481	0.517	0.288
Upward, strength 15	E	2.04	1.04	3.02	0.505	0.537	0.877
HW							
$(3.2, 4^\circ, 2^\circ)$	I	1.73	0.968	1.22	0.676	0.519	0.0980
$(3.4, 4^\circ, 3^\circ)$	I	1.62	1.07	1.28	0.729	0.506	0.492
$(3.6, 3^\circ, 1^\circ)$	I	1.76	1.46	1.46	0.815	0.570	0.104
$(3.2, 4^\circ, 4^\circ)$	E	1.47	0.945	1.11	0.688	0.502	0.122
$(4.0, 0^\circ, 1^\circ)$	E	0.881	0.767	0.752	0.416	0.359	0.266
$(4.8, 4^\circ, 3^\circ)$	E	0.849	0.719	0.812	0.375	0.364	0.127
X38							
$\alpha = 22.5^\circ$	I	1.08	0.785	0.897	0.490	0.704	0.123
$\alpha = 28.5^\circ$	I	1.02	0.731	0.911	0.466	0.680	0.111
$\alpha = 34.5^\circ$	I	1.02	0.742	0.873	0.460	0.677	0.116
$\alpha = 36.5^\circ$	I	1.03	0.752	0.879	0.466	0.681	0.119
$\alpha = 20^\circ$	E	1.14	0.839	0.873	0.503	0.723	0.143
$\alpha = 40^\circ$	E	1.05	0.788	0.889	0.474	0.692	0.138

DAFL achieves the lowest full field RL2 in 17 of the 18 test cases (Table 8). These cases include both interpolation and extrapolation settings. The exception is the CHN-T2 extrapolation case with upward deflection at strength 15. In this case, warm-starting with DAFL still requires fewer solver iterations than cold-starting or any convergent baseline (Figure 14).

D.2 ACCURACY ACROSS PHYSICAL VARIABLES

The models predict u, v, w, ρ, p, k, ω, and μ_t for CHN-T2 and X38. HW also includes temperature T. Table 9 reports the mean RL2 for each variable over six test cases per dataset. The symbol "–" indicates that T is not predicted for CHN-T2 or X38.

Table 9: RL2 for each output variable (↓). Values are averaged over six test cases per dataset.

Method	u	v	w	ρ	p	T	k	ω	μ_t
CHN-T2									
LNO	0.325	0.959	1.18	0.134	0.112	–	4.68	2.52	1.96
GINO	0.328	1.59	1.51	0.115	0.120	–	1.21	1.15	1.04
Transolver-3	0.339	8.30	6.47	0.692	0.785	–	4.09	1.66	1.92
AB-UPT	0.155	0.612	0.658	0.0421	0.0438	–	0.742	0.979	0.720
AdaField	0.139	0.905	0.932	0.0682	0.0837	–	0.688	0.510	0.910
DAFL	0.0348	0.196	0.237	0.0128	0.0147	–	1.13	0.307	0.942

Table 9 (continued)

Method	u	v	w	ρ	p	T	k	ω	μ_t
HW									
LNO	0.503	1.02	1.03	0.560	0.738	0.522	1.02	1.11	5.96
GINO	0.415	0.878	0.909	0.486	0.651	0.447	0.940	0.998	3.16
Transolver-3	0.408	0.876	0.901	0.561	0.867	0.429	0.670	1.02	4.21
AB-UPT	0.0943	0.302	0.278	0.155	0.101	0.117	0.348	0.946	3.21
AdaField	0.145	0.393	0.300	0.123	0.237	0.158	0.395	0.320	2.16
DAFL	0.0257	0.140	0.132	0.0395	0.0487	0.0441	0.228	0.214	0.942
X38									
LNO	0.413	0.466	1.00	0.660	0.928	–	0.987	1.12	1.08
GINO	0.251	0.288	0.809	0.368	0.525	–	0.870	1.05	0.766
Transolver-3	0.304	0.456	1.30	0.655	0.876	–	0.976	1.41	1.13
AB-UPT	0.0993	0.159	0.476	0.165	0.163	–	0.696	1.12	0.936
AdaField	0.191	0.435	1.00	0.492	0.600	–	0.852	0.983	0.988
DAFL	0.0174	0.0315	0.112	0.0528	0.0628	–	0.233	0.248	0.243

On HW and X38, DAFL achieves the lowest RL2 for every output variable. Relative to the best baseline for each variable, it reduces RL2 by 33%–73% on HW and 61%–83% on X38. These reductions indicate substantial gains in prediction accuracy. On CHN-T2, DAFL achieves the lowest RL2 for six of the eight variables. For these six variables, the reductions are about 40%–75% relative to the best baseline. The exceptions are k and μ_t.

Tables 10 and 11 report MSE and MAE for each variable. MSE gives greater weight to large errors, while MAE reports the average absolute error across all points.

Table 10: MSE for each output variable (↓). Values are averaged over six test cases per dataset.

Method	u	v	w	ρ	p	T	k	ω	μ_t
CHN-T2									
LNO	0.0614	0.00480	0.00568	0.0174	0.00653	–	1.82×10^{-5}	4.60×10^{-4}	1.05×10^{7}
GINO	0.0625	0.0132	0.00924	0.0129	0.00748	–	1.25×10^{-6}	9.58×10^{-5}	4.00×10^{6}
Transolver-3	0.0667	0.359	0.170	0.466	0.322	–	1.40×10^{-5}	2.01×10^{-4}	9.90×10^{6}
AB-UPT	0.0139	0.00195	0.00177	0.00173	0.00100	–	4.97×10^{-7}	6.94×10^{-5}	2.40×10^{6}
AdaField	0.0113	0.00427	0.00354	0.00453	0.00366	–	4.41×10^{-7}	1.89×10^{-5}	3.37×10^{6}
DAFL	7.36×10^{-4}	2.02×10^{-4}	2.34×10^{-4}	1.62×10^{-4}	1.16×10^{-4}	–	1.95×10^{-6}	7.01×10^{-6}	4.21×10^{6}
HW									
LNO	2.19	0.224	0.223	0.513	1.83	1.17	0.0342	300	9.89×10^{3}
GINO	1.49	0.165	0.172	0.389	1.44	0.865	0.0295	244	4.28×10^{3}
Transolver-3	1.51	0.184	0.181	0.509	2.22	0.762	0.0155	256	9.21×10^{3}
AB-UPT	0.0773	0.0200	0.0167	0.0430	0.0350	0.0601	0.00451	219	6.90×10^{3}
AdaField	0.222	0.0352	0.0201	0.0272	0.220	0.121	0.00540	25.1	7.37×10^{3}
DAFL	0.00609	0.00477	0.00393	0.00286	0.00990	0.00900	0.00211	11.4	2.46×10^{3}
X38									
LNO	3.35	1.47	0.369	0.648	13.2	–	3.09×10^{-4}	60.0	554
GINO	1.24	0.549	0.238	0.200	4.18	–	2.40×10^{-4}	52.2	283
Transolver-3	1.79	1.39	0.615	0.643	11.9	–	3.03×10^{-4}	94.0	599
AB-UPT	0.193	0.156	0.0822	0.0403	0.417	–	1.53×10^{-4}	59.0	416
AdaField	0.715	1.09	0.369	0.365	5.66	–	2.30×10^{-4}	45.7	463
DAFL	0.00601	0.00610	0.00458	0.00425	0.0622	–	1.73×10^{-5}	2.92	28.6

Table 11: MAE for each output variable (↓). Values are averaged over six test cases per dataset.

Method	u	v	w	ρ	p	T	k	ω	μ_t
CHN-T2									
LNO	0.182	0.0444	0.0485	0.119	0.0580	–	0.00414	0.0185	2.62×10^3
GINO	0.189	0.0925	0.0693	0.0779	0.0589	–	7.14×10^{-4}	0.00337	811
Transolver-3	0.151	0.333	0.236	0.382	0.326	–	0.00206	0.00702	2.00×10^3
AB-UPT	0.0672	0.0224	0.0211	0.0213	0.0149	–	3.44×10^{-4}	0.00227	593
AdaField	0.0520	0.0375	0.0312	0.0314	0.0291	–	2.29×10^{-4}	7.03×10^{-4}	643
DAFL	0.00636	0.00390	0.00389	0.00355	0.00306	–	8.15×10^{-5}	1.69×10^{-4}	110
HW									
LNO	1.18	0.322	0.318	0.429	0.642	0.724	0.0791	7.25	33.4
GINO	0.942	0.254	0.264	0.407	0.571	0.582	0.0647	3.65	17.7
Transolver-3	0.883	0.246	0.263	0.407	0.667	0.512	0.0363	3.88	31.1
AB-UPT	0.147	0.0882	0.0820	0.111	0.112	0.158	0.0305	3.63	18.2
AdaField	0.376	0.135	0.0958	0.0942	0.186	0.190	0.0203	0.781	10.9
DAFL	0.0437	0.0329	0.0305	0.0228	0.0289	0.0391	0.00720	0.178	3.06
X38									
LNO	1.26	1.02	0.369	0.532	2.07	–	0.00522	2.84	12.5
GINO	0.659	0.440	0.302	0.249	0.762	–	0.00568	2.75	6.10
Transolver-3	0.891	0.764	0.464	0.422	1.30	–	0.00718	3.45	12.8
AB-UPT	0.237	0.257	0.187	0.131	0.424	–	0.00493	3.27	7.44
AdaField	0.572	0.805	0.375	0.357	1.21	–	0.00712	3.24	5.56
DAFL	0.0354	0.0370	0.0313	0.0238	0.0658	–	7.24×10^{-4}	0.101	0.631

On HW and X38, DAFL also achieves the lowest MSE and MAE for every output variable. On CHN-T2, it achieves the lowest MAE for all eight variables and the lowest MSE for six. For k and μ_t, DAFL's MSE and RL2 are higher than the best baseline values despite its lower MAE. Their distributions are highly imbalanced (Figure 11). This difference may reflect large local errors at a small number of points. Such errors have a stronger effect on MSE and RL2 than on MAE. The results for individual cases in Table 8 indicate that DAFL retains its overall full field accuracy advantage despite these exceptions.

D.3 SPATIAL STRUCTURES AND ERROR PATTERNS

Figures 12 and 13 compare DAFL with all five baselines for the representative interpolation and extrapolation cases in Figure 4. The first figure shows Mach number, and the second shows turbulent kinetic energy k. The visualized variables are defined in Appendix A.3.

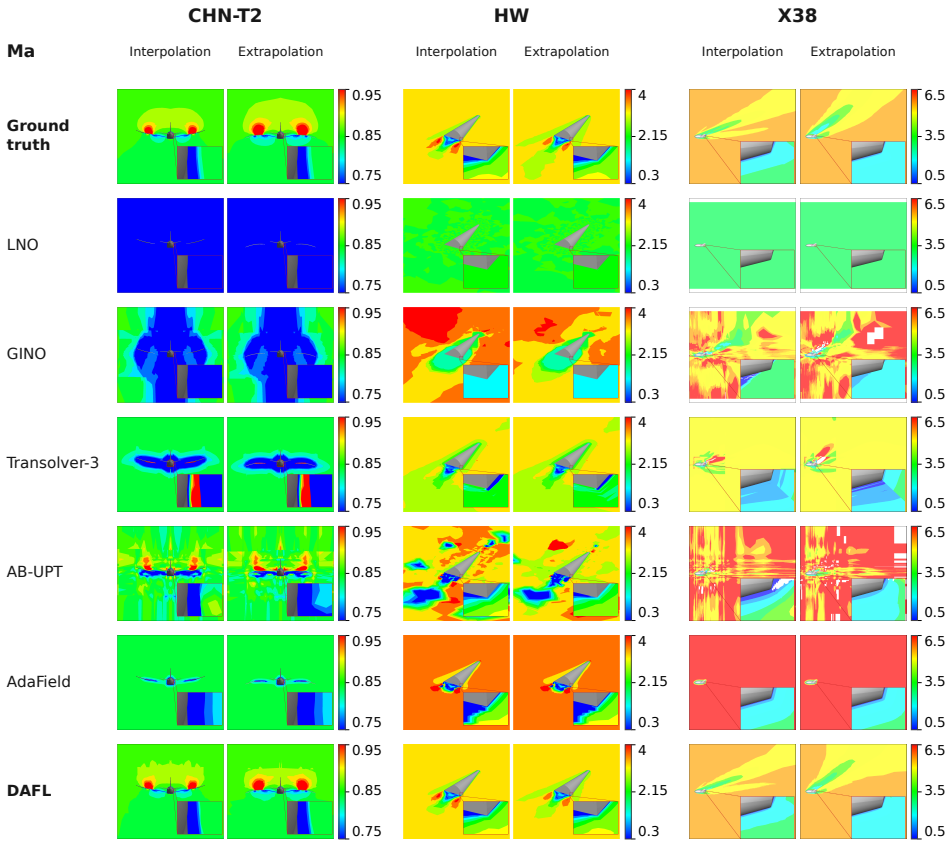


Figure 12: Mach number fields for representative interpolation and extrapolation cases on CHN-T2, HW, and X38. For each dataset, the left column shows interpolation and the right column shows extrapolation. Rows show the ground truth and predictions from each method. Insets show regions near the body. White regions in predicted fields indicate locations where Mach number cannot be computed.

The predictions of the five baselines show errors in the background Mach number or local flow structures. GINO and AB-UPT also produce banded or fragmented patterns. On X38, both methods produce white regions where Mach number cannot be computed. Mach number fields derived from DAFL's predicted flow variables more closely match the ground truth across all three datasets. On CHN-T2, they capture the regions of high Mach number near both wings. On HW, they recover the background Mach number and the flow structures near the body and downstream. On X38, they preserve the flow structures downstream of the aircraft. The insets further show that the DAFL predictions better resolve local Mach number variations near the wall. These visual observations are consistent with DAFL's lower full field RL2 values in Table 1.

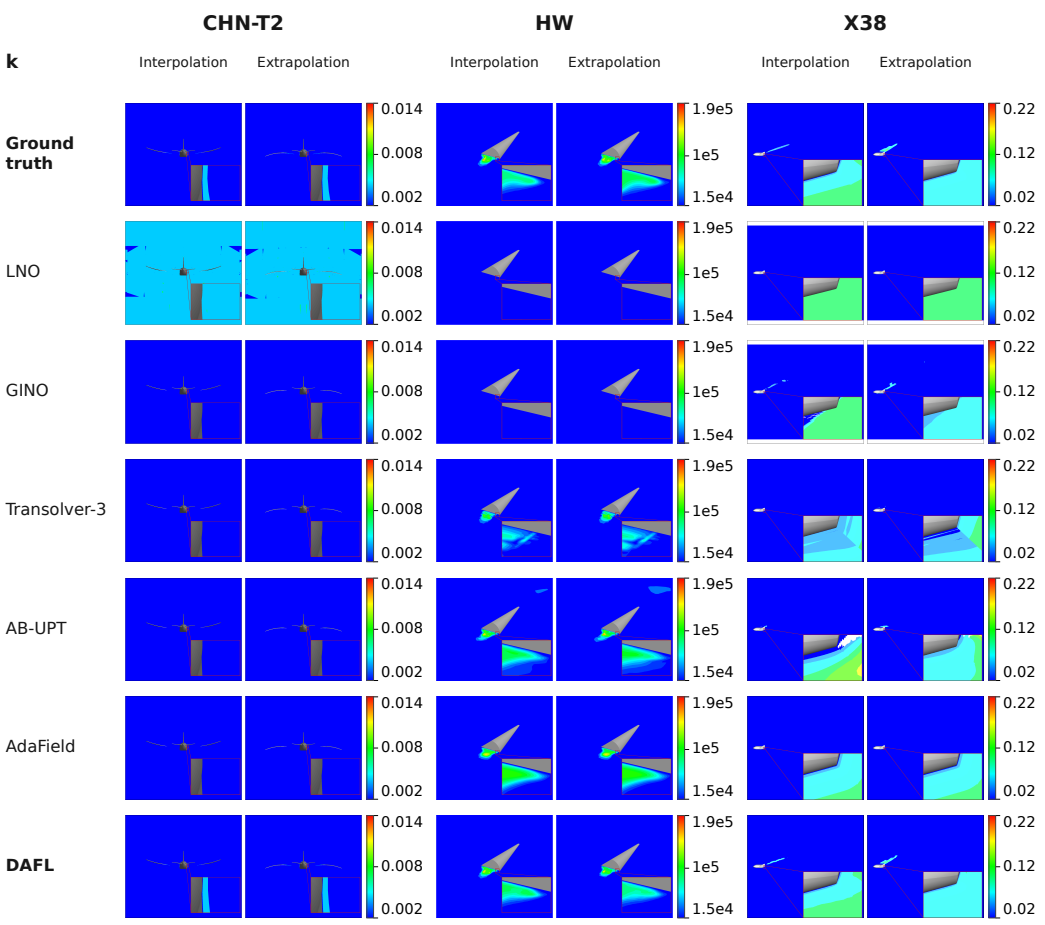


Figure 13: Turbulent kinetic energy k for representative interpolation and extrapolation cases on CHN-T2, HW, and X38. Rows show the ground truth and predictions from each method.

Figure 13 shows that regions of high k are spatially localized. On HW, LNO and GINO miss the downstream region of high k, while AB-UPT overestimates its extent. The DAFL predictions better preserve the location and spatial extent of these regions across all three datasets.

E CFD Warm-Starting Protocol and Full Results

E.1 Numerical solver and initial states

All CFD simulations use the CFD module of NNW-FSI, a fluid-structure interaction software package within the National Numerical Wind Tunnel (NNW) platform. The platform was developed by the China Aerodynamics Research and Development Center (CARDC). The module solves the Reynolds-averaged Navier–Stokes (RANS) equations using the finite-volume method. Table 12 summarizes the numerical settings used for all three datasets. CFD warm-starting begins from a predicted initial state, whereas cold-starting begins from a uniform initial state. For each test case, both approaches use identical numerical settings, iteration limits, and convergence criteria. This setup isolates the effect of the initial state on convergence.

Table 12: NNW-FSI numerical settings shared across the three datasets.

Item	Setting
Solver and equations	NNW-FSI, RANS
Spatial discretization	Finite-volume method
Turbulence model	Two-equation SST
Inviscid flux	MUSCL–Roe
Viscous/diffusion terms	Second-order central scheme
Iterative solver	Unrelaxed LU-SGS, geometric multigrid with three levels
Gas model	Calorically perfect air, $\gamma = 1.4$, $R = 287\ \mathrm{J/(kg \cdot K)}$
Prandtl numbers	0.72 (molecular), 0.9 (turbulent)
Maximum iterations	10,000

E.2 Convergence criterion and iteration speedup

The same convergence criterion is applied to all CFD simulations. For each test case, C_{ref} denotes the drag coefficient of the converged cold-start solution. In each simulation, N_{end} is the final recorded iteration, and $C^{(n)}$ is the drag coefficient at iteration n, where $1 \le n \le N_{\mathrm{end}}$. The absolute tolerance is $\varepsilon = 2.5 \times 10^{-4}$. The convergence iteration is defined as

$$N_{\mathrm{conv}} = \min \left\{ n : \max_{n \le j \le N_{\mathrm{end}}} |C^{(j)} - C_{\mathrm{ref}}| \le \varepsilon \right\}. \tag{20}$$

Thus, N_{conv} is the earliest iteration from which the drag coefficient remains within $C_{\mathrm{ref}} \pm \varepsilon$ through iteration N_{end}. A simulation is unsuccessful if the set is empty or the solver terminates abnormally. In Eq. equation 10, N_{cold} and N_{warm} denote the convergence iterations for cold-starting and warm-starting, respectively.

E.3 CFD warm-starting and iteration speedup across test cases

Figure 14 shows iteration speedups and convergence outcomes for all 18 test cases, including interpolation and extrapolation cases in each dataset.

DAFL achieves convergence under CFD warm-starting in all 18 test cases. In each case, it requires fewer solver iterations than cold-starting and every baseline that converges. Its speedup ranges from $1.25\times$ on HW to $12.88\times$ on X38.

On CHN-T2, AB-UPT and AdaField converge in all six cases but require more iterations than cold-starting. LNO, GINO, and Transolver-3 do not converge in any CHN-T2 case. On HW, GINO and Transolver-3 reduce the iteration count in four cases each. Both fail to converge at the interpolation condition $(3.2, 4^\circ, 2^\circ)$ and the extrapolation condition $(4.8, 4^\circ, 3^\circ)$. LNO reduces the iteration count in five cases but requires more iterations than cold-starting at $(4.8, 4^\circ, 3^\circ)$. AdaField reduces the iteration count in all six HW cases. AB-UPT does not converge in any HW case. On X38, none of the five baselines converges.

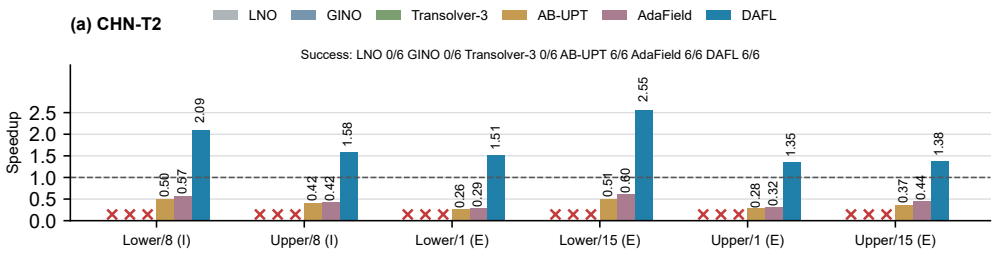


Figure 14: CFD iteration speedup on CHN-T2. The dashed line marks cold-start parity; red crosses mark non-convergence. HW and X38 continue on the next page.

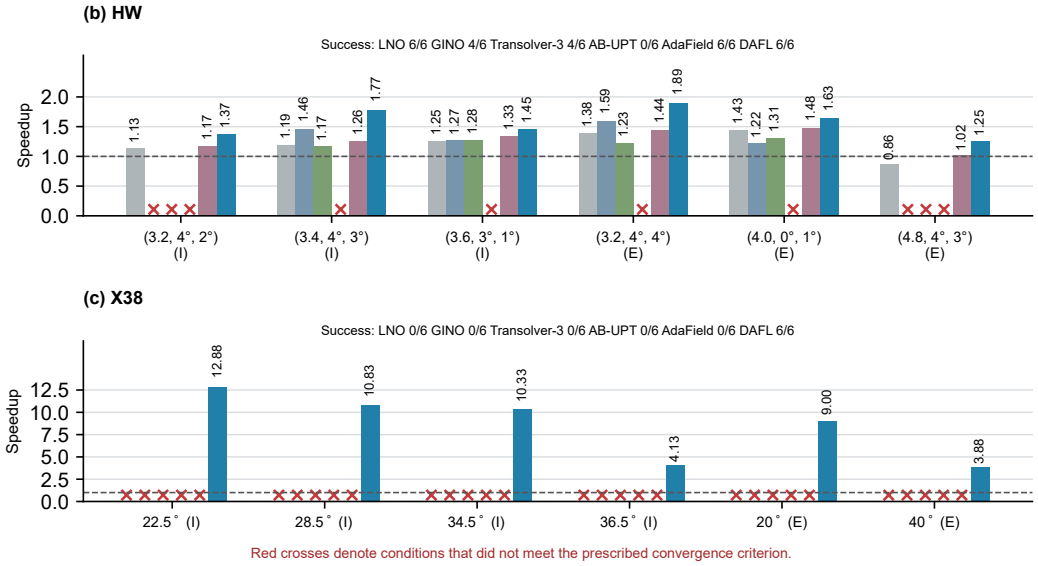


Figure 14: Continued. CFD iteration speedup for (b) HW and (c) X38. The dashed line at 1 indicates the same number of iterations as cold-starting. Bars above the line indicate fewer iterations, and bars below indicate more. Red crosses mark simulations that did not converge.

F Supplementary Ablation Study

F.1 Accuracy across test cases

Table 13: Full field RL2 (↓) for DAFL and five ablation variants on X38. Each ablation row identifies the modified component. Columns give the angle of attack α. The first four cases are interpolation, and the last two are extrapolation.

Variant	22.5°	28.5°	34.5°	36.5°	20°	40°
DAFL (full)	0.123	0.111	0.116	0.119	0.143	0.138
Voxel sampling	0.165	0.152	0.154	0.159	0.190	0.177
Disturbance-aware sampling and regression	2.633	2.601	2.331	2.235	2.585	2.103
Logarithmic wall distance	0.431	0.432	0.438	0.440	0.441	0.442
Log-space regression	0.267	0.265	0.270	0.277	0.278	0.290
Multistage modulation by geometry and flow conditions	0.323	0.319	0.322	0.330	0.338	0.353

All five ablations increase full field RL2 in every X38 test case. Removing disturbance-aware sampling and regression produces the largest increase. This variant uses a single voxel size and regression head for both weak and strong disturbance regions. The large increase in RL2 indicates that a single sampling scale and regression head are less effective at modeling critical disturbances.

F.2 ACCURACY ACROSS PHYSICAL VARIABLES

Table 14: RL2 for each output variable (↓), comparing DAFL with five ablation variants on X38.

Variant	ρ	u	v	w	p	k	ω	μ_t
DAFL (full)	0.0528	0.0174	0.0315	0.112	0.0628	0.233	0.248	0.243
Voxel sampling	0.0906	0.0291	0.0497	0.185	0.110	0.275	0.237	0.353
Disturbance-aware sampling and regression	0.413	0.0996	0.149	0.499	0.679	1.78	14.6	1.06
Logarithmic wall distance	0.272	0.195	0.192	0.357	0.447	0.653	1.02	0.361
Log-space regression	0.225	0.0422	0.0743	0.245	0.380	0.398	0.516	0.315
Multistage modulation by geometry and flow conditions	0.307	0.0483	0.0759	0.242	0.520	0.584	0.495	0.373

Replacing voxel sampling with random point sampling slightly reduces ω RL2 from 0.248 to 0.237. RL2 increases for the other seven variables. Random point sampling assigns greater sampling probability to areas with higher point density. Voxel sampling provides more uniform spatial coverage.

Removing disturbance-aware sampling and regression raises ω RL2 from 0.248 to 14.6, the largest relative increase among the eight variables. The near and outer regions have different ω distributions (Figure 9). This result supports adapting both sampling and regression to disturbance regions.

Removing logarithmic wall distance raises u RL2 from 0.0174 to 0.195, the largest relative increase among the eight variables. The corresponding Mach number field shows larger errors near the wall (Figure 7). These results support using logarithmic wall distance to represent near-wall flow variation.

Removing log-space regression and the exponential transformation increases RL2 for all eight variables. For k and ω, the RL2 ratios relative to DAFL are 1.71 and 2.08, respectively. These results support log-space regression for variables with imbalanced distributions and the exponential transformation to preserve physically valid predictions (Figure 7).

Restricting modulation by geometry and flow conditions to the first encoder stage increases RL2 for all eight variables. These results support injecting geometry and flow conditions throughout the backbone to capture case-dependent variations.