LEARNING TO DECOMPOSE PDE DOMAINS WITH GEOMETRY-AWARE LOCAL OPERATORS AND GLOBAL COMMUNICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural operators provide an efficient approach to repeated partial differential equation (PDE) solving, but practical use on complex geometries and large meshes remains constrained by the challenge of balancing local spatial resolution, global physical coupling, and computational cost. A Geometry-Aware Domain-Decomposition Neural Operator (GDDNO) is introduced to organize local computation and global communication through domain decomposition. Within each subdomain, local interactions are encoded by a shared graph operator. The global information is exchanged between the subdomain tokens through attention, and the resulting context is combined with retained cell features for prediction. Overlap domain assembly and flux consistency constraints coordinate predictions between subdomains. GDDNO is evaluated on five RealPDEBench systems, AirfRANS, and a three-dimensional reacting flow dataset. It obtains the lowest simulation-trained relative L_2 error on all five RealPDEBench systems. On full-field AirfRANS, the relative L_2 error is 0.0815, compared with 0.3511 for the strongest baseline, and the prediction of the aerodynamic coefficients is also the best. On cavity combustion flow, the relative L_2 errors for the predicted fields and combustion efficiency are 0.4148 and 0.2112, respectively, both the lowest among the compared models. These results indicate that domain decomposition can be used as an effective strategy to combine local geometric detail with long-range interactions in neural PDE operators.

1 INTRODUCTION

Many scientific and engineering workflows require the repeated solution of partial differential equations(PDE) under changing geometries, boundary conditions, and physical parameters. Conventional PDE numerical solvers provide reliable solutions but are often too expensive for design optimization, uncertainty analysis, inverse problems, and real-time control. Neural operators reduce this cost by learning mappings between function spaces and amortizing the solution process across a family of PDE instances (Li et al., 2020b; Lu et al., 2021). Their contribution is that the same trained model can be directly evaluated on many new inputs without solving new optimization problems for each instance.

For operator learning, especially in the case of large meshes, the organization of computation is important. Spectral-based methods such as the FNO provide efficient global mixing on regular grids (Li et al., 2021), and geometric extensions map irregular domains to latent representations (Li et al., 2023a). Graph models retain mesh connectivity (Li et al., 2026), and describe local interactions directly (Pfaff et al., 2020). In recent years, Transformers (Vaswani et al., 2017) have confirmed their excellent performance on physical problems, and attention plays a surprising role in deep physical feature learning (Hao et al., 2023; Huang et al., 2026; Li et al., 2022; Zuo et al., 2023). Global attention can represent long-range physical dependence, but direct attention over millions of mesh cells incurs quadratic cost. Random subsampling or independent patch processing reduces the immediate memory demand, but may lose fine local structures or introduce artificial boundaries. This raises a design question about how a large physical domain should be represented and coupled during operator learning.

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.

The same local-to-global challenge has long been addressed in computational fluid dynamics(CFD). Domain decomposition provides a useful computational pattern. Parallel numerical solvers divide a global domain into local problems, exchange information at subdomain boundaries, and assemble a global solution (Chan & Mathew, 1994; Toselli & Widlund, 2005). The partition is therefore more than a approach of reducing the size of each calculation. It specifies where information is available and where compatibility must be maintained. This computational organization motivates an architecture that jointly learns local interactions and global communication.

GDDNO uses this pattern to combine local mesh computation with global communication. Nonoverlapping core regions are expanded with halo cells. A shared graph operator encodes each expanded subdomain from cell features and face geometry. Learned tokens compress the local features, and attention exchanges information across all subdomains. The global context is returned to cells before decoding. Predictions for shared cells are assembled by global cell identity, with training losses that reduce disagreement in overlap regions and across core boundaries.

We evaluate GDDNO on benchmark datasets including structured spatiotemporal systems, irregular airfoil flows, and three-dimensional reacting flows on million-scale grids, GDDNO achieved excellent results in all dataset. The main contributions are summarized as:

- A neural operator is constructed around overlapping mesh subdomains, geometry-aware local graph computation, compressed global communication, and shared-cell assembly.
- Local cell features are retained while subdomain-tokens carry long-range information. Overlap and interface flux losses coordinate predictions near artificial boundaries.
- GDDNO is evaluated on five RealPDEBench systems, full-field and near-wall AirfRANS tasks, and a three-dimensional reactive flow dataset. The evidence includes field accuracy, engineering quantities, and simulation-to-real transfer.

2 Related work

2.1 Physics-informed learning and neural operators

Since Raissi et al. (2019) proposed physics-informed neural networks (PINNs), which exploit the universal approximation property of neural networks (Hornik et al., 1989), solving PDE via neural networks has received substantial research interest. This method established a general way to place governing equations inside a learning objective and has been extended to inverse problems, multiphysics systems, and noisy or incomplete observations. In standard PINN formulations, a separate optimization problem is solved for each new physical instance (Karniadakis et al., 2021). Neural operators learn mappings between function spaces across parameterized families of PDEs, allowing a trained model to be reused for new inputs (Kovachki et al., 2023). Physical equations can also be incorporated into operator training through residual constraints, reducing dependence on labeled data while retaining efficient inference ability(Li et al., 2024), and therefore has received more attention in recent years.

2.2 Local and global computation in neural operators

For operator learning, architectural choices determine how local spatial detail and long-range dependence are represented. FNO performs global mixing in Fourier space (Li et al., 2021), while multiscale representations are used in MWT and WDNO (Hu et al., 2025; Gupta et al., 2021). On irregular meshes, connectivity and relative geometry provide a direct representation of local interactions. MeshGraphNets and message-passing PDE solvers use mesh connectivity to propagate information between neighboring elements (Pfaff et al., 2020; Brandstetter et al., 2021). The spatial reach of local message passing increases with network depth. Hierarchical interactions, as used in MGKN, provide another route to longer-range coupling (Li et al., 2020b).

Global computation can also be performed on representations smaller than the original discretization. Geo-FNO learns a deformation to a regular latent grid (Li et al., 2023a). GINO connects irregular geometry to a latent Fourier grid through a graph (Li et al., 2023b). Transformer-based operators also use compact representations to reduce the cost of global interaction (Wu et al., 2024; Alkin et al., 2024; Hu et al., 2026b). These methods advance local modeling and global interaction, but their integration with cell-level feature retention remains an architectural design challenge.

2.3 Domain decomposition with learned solvers

Classical parallel computing method in finite element analysis or computational fluid dynamics organize a global solve through local solution, information exchange, and assembly (Chan & Mathew, 1994; Toselli & Widlund, 2005). Its effectiveness depends on both the local solver and the coupling between subdomains. This perspective has motivated learning within domain decomposition, including learned coarse spaces and preconditioners Heinlein et al. (2021).

In physics-informed approaches, local networks can be coupled through boundary or interface terms (Li et al., 2020a; Jagtap et al., 2020). XPINNs support spatial and temporal partitions (Jagtap & Karniadakis, 2020)., while FBPINNs combine basis functions supported on overlapping subdomains (Moseley et al., 2023). Decomposition has also been incorporated into the construction and evaluation of neural operators. NUNO constructs locally regular grids to control interpolation error (Liu et al., 2023), and GNOT uses geometric gating for soft domain decomposition (Hao et al., 2023). Schwarz Neural Inference couples trained local operators through iterative Schwarz updates at inference time (Huang et al., 2026). These approaches establish decomposition as a useful modeling strategy, but combining geometric fidelity, global coupling, and interface coordination without iterative subdomain solves requires further study.

3 Methodology

3.1 Problem setup

Consider a family of PDEs defined on a physical domain $\Omega(a)$. The input a specifies the geometry, coefficient fields, boundary conditions, and operating parameters. The solution field u satisfies

$$\mathcal{N}(u; a) = 0 \quad \text{in } \Omega(a), \qquad \mathcal{B}(u; a) = 0 \quad \text{on } \partial\Omega(a),$$

where $\mathcal{N}$ and $\mathcal{B}$ denote the governing equations and boundary conditions. For unsteady problems, the input also includes initial conditions and observed states. The target is the subsequent solution over the prescribed prediction interval. Operator learning approximates the solution map $\mathcal{G}^\dagger : a \mapsto u$ from paired inputs and solutions.

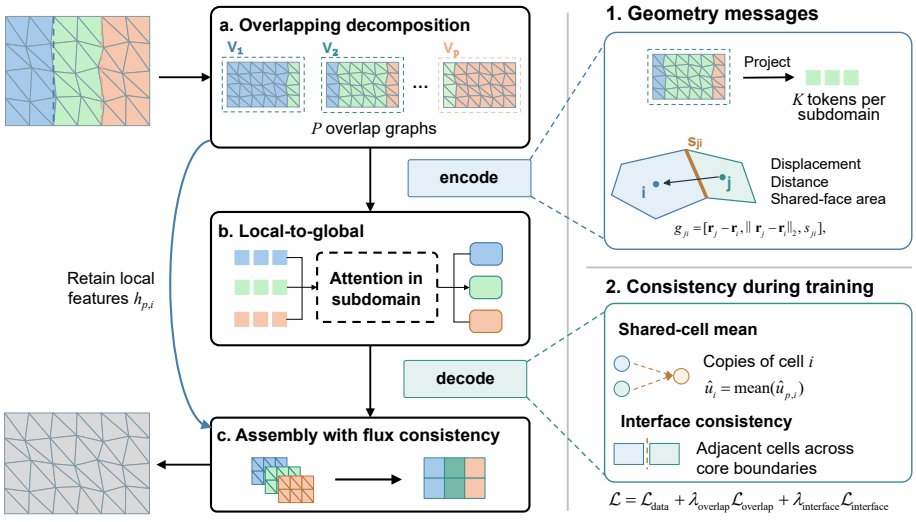

Figure 1: GDDNO combines geometry-aware local encoding on overlapping subdomains with global communication through compact tokens. Retained local features and global context guide decoding, while shared cell averaging and flux consistency constraints coordinate predictions into a single global field.

3.2 FROM PARALLEL CFD TO A LEARNED DECOMPOSITION

In a parallel finite-volume solver, cells are assigned to subdomains and information is exchanged across their artificial boundaries. For a conserved quantity, a shared-face flux must enter neighboring control-volume balances with opposite signs:

$$F_{ij} + F_{ji} = 0,$$

where F_{ij} is the outward numerical flux from cell i through its face with cell j. This cancellation must also hold when the cells belong to different processors. Each subdomain stores copies of neighboring cells, called halo cells. These copies are updated through communication between processors to provide the physical quantities required for interface flux calculations.

GDDNO adopts these computational roles within a learned operator. The mesh is first partitioned into P nonoverlapping core regions. Each core is expanded by a fixed number of adjacency hops to form an overlapping cell set V_p. A shared graph network encodes the cells in V_p. Learned tokens exchange information among subdomains, and the decoded predictions are assembled by global cell identity. Two consistency losses constrain predictions for shared cells and neighboring states across core boundaries. The mesh is partitioned in advance, and each local cell is matched to its corresponding cell in the global mesh. At inference, a single forward pass produces the assembled global field. Figure 1 illustrates this correspondence. Section 3.3 and 3.4 specify the predictive map on each subdomain, and Section 3.5 gives its assembly and training objective.

3.3 GEOMETRY-AWARE LOCAL COMPUTATION

Each expanded subdomain is represented as a cell graph. Within each subdomain, the graph encoder conditions neighboring interactions on their face geometry. For a directed edge from cell j to cell i, its geometric features are:

$$g_{ji} = [\boldsymbol{r}_j - \boldsymbol{r}_i,\ \|\boldsymbol{r}_j - \boldsymbol{r}_i\|_2,\ s_{ji}]$$

where $\boldsymbol{r}_i$ is the cell centroid and s_{ji} is the shared-face area. Relative displacement and distance describe the spatial relation, while face area supplies information about the finite-volume interaction. A linear projection of x_i initializes the hidden feature $h_{p,i}$ in each subdomain. One graph layer computes:

$$\overline{m}_{p,i} = \frac{1}{|\mathcal{N}_p(i)|} \sum_{j \in \mathcal{N}_p(i)} \phi_{\mathrm{msg}}([h_{p,j}, g_{ji}])$$

$$h_{p,i}^{\mathrm{new}} = \mathrm{LN}(h_{p,i} + \phi_{\mathrm{upd}}([h_{p,i}, \overline{m}_{p,i}]))$$

where $\mathcal{N}_p(i)$ contains the incoming neighbors of cell i in V_p, and LN is layer normalization. The message and update mappings ϕ_{msg} and ϕ_{upd} are multilayer perceptrons. Each layer has its own parameters, shared across subdomains.

3.4 GLOBAL COMMUNICATION THROUGH SUBDOMAIN-TOKENS

Subdoamin-token. The local features are compressed into K tokens per subdomain using learned queries t_k.

$$\alpha_{p,k,i} = \mathrm{softmax}_{i \in V_p}\left(\frac{t_k^\top W_{\mathrm{key}} h_{p,i}}{\sqrt{d}}\right) \qquad z_{p,k} = W_{\mathrm{out}} \sum_{i \in V_p} \alpha_{p,k,i} W_{\mathrm{value}} h_{p,i}$$

where d is the hidden width and the W mappings are learned linear projections. The softmax is taken over cells within one subdomain. Each query defines a separate weighted summary, allowing multiple physical structures within a subdomain to contribute differently to communication. The queries and projections are shared across subdomains and trained with the local operator.

Local-to-global communication. All PK tokens form one sequence. For one attention head, the interaction from token (q, l) to token (p, k) is weighted by:

$$A_{p,k;q,l} = \mathrm{softmax}_{q,l}\left(\frac{(W_Q z_{p,k})^\top (W_K z_{q,l})}{\sqrt{d}}\right)$$

Attention is evaluated over all subdomains, allowing distant inputs to affect the context returned to a local region. The pairwise attention cost is $\mathcal{O}((PK)^2 d)$, compared with $\mathcal{O}(N^2 d)$ for full-mesh attention.

Return to local cells. Let $\widetilde{z}_{p,k}$ denote the final communicated tokens. Their subdomain mean is fused with the retained local features.

$$\overline{z}_p = \frac{1}{K} \sum_{k=1}^{K} \widetilde{z}_{p,k} \qquad \widehat{u}_{p,i} = \phi_{\mathrm{dec}}([h_{p,i}, \overline{z}_p])$$

The shared decoder ϕ_{dec} includes feature fusion and output projection. Global information is broadcast through $\overline{z}_p$, while spatial variation within the subdomain remains available through $h_{p,i}$.

3.5 Assembly and interface coordination

Expanded subdomains produce multiple predictions near artificial boundaries. Let $\mathcal{C}(i)$ be the set of subdomains containing global cell i, with $m_i = |\mathcal{C}(i)|$. The assembled prediction is:

$$\widehat{u}_i = \frac{1}{m_i} \sum_{p \in \mathcal{C}(i)} \widehat{u}_{p,i}$$

Global cell identifiers make this operation exact at the level of cell correspondence. Each physical cell is supervised once, regardless of how many overlap copies are stored. In order to avoid the inconsistency of the same cell in different subdomain-tokens, qdditional training constraints penalize inconsistent predictions for the same cell across overlapping subdomains:

$$\mathcal{L} = \mathcal{L}_{\mathrm{data}} + \lambda_{\mathrm{overlap}} \mathcal{L}_{\mathrm{overlap}} + \lambda_{\mathrm{interface}} \mathcal{L}_{\mathrm{interface}}$$

The data term measures the mean squared error of the complete field. The overlap term penalizes differences between each shared-cell prediction and its assembled mean. The interface term penalizes differences between neighboring predicted states on opposite sides of a core boundary. The first constraint compares copies of the same physical cell, while the second coordinates the two sides of an interface.

4 Experiment

GDDNO was evaluated on three benchmark groups with different geometry, discretization, and physical complexity. AirfRANS contains high-fidelity steady Reynolds-averaged Navier-Stokes solutions around airfoils with variations in geometry, Reynolds number, and angle of attack (Bonnet et al., 2022). RealPDEBench contains paired numerical and experimental observations for five spatialtemporal systems and is designed to evaluate simulation pretraining followed by real-world fine-tuning (Hu et al., 2026a). The cavity combustion flow dataset is a benchmark for reacting flow , including the velocity, temperature, pressure, and reaction species involved in the reacting flow simulation which further tests the operator's ability to model multiphysics coupling phenomena.

Table 1: Benchmarks used in this study.

Dataset	**Physical systems**	**Spatial representation**
AirfRANS	Steady incompressible airfoil RANS	Irregular CFD cell graph
RealPDEBench	Cylinder, controlled cylinder, fluid-structure interaction, foil, combustion	Structured spatiotemporal measurements and simulations
Cavity combustion flow	Three-dimensional reacting flow	Unstructured finite-volume mesh

The baselines include branch–trunk, spectral, convolutional, wavelet, diffusion, and attention-based architectures, with grid, graph, or point-set representations. These operators include U-Net (Ronneberger et al., 2015), GraphSAGE (Hamilton et al., 2017), PointNet (Charles et al., 2017), DeepONet (Lu et al., 2021), FNO (Li et al., 2021), CNO (Raonic et al., 2023), MWT (Gupta et al., 2021),

WDNO (Hu et al., 2025), Galerkin Transformer (Cao, 2021), and Transolver (Wu et al., 2024). To ensure fairness as much as possible. every comparison within a benchmark uses the same data split, target fields, normalization, and training hyperparameters. All models were optimized using Adam with an initial learning rate of 0.0001 and a OneCycleLR schedule.

The main metrics are root mean squared error, relative L_2 error, and R^2 of predicted fields. AirfRANS also reports the relative L_2 error of the wall pressure-coefficient distribution C_p and lift coefficient C_l. RealPDEBench uses the Kinetic Energy Error (KE), Frequency Error (FE), Fourier-space Error (fRMSE) and other metrics defined in the RealPDEBench study. Cavity combustion flow provides the combustion efficiency (η) that researchers are most concerned about in reacting flow simulation, all flow parameters and reaction species are used in the calculation of this metrics. See Appendix B for comprehensive descriptions of implementations.

4.1 Main Result

4.1.1 AirfRANS

Airfoil design requires a large number of simulations of different geometries, incoming flow velocities, angles of attack, and Reynolds numbers, making operator learning particularly suitable for this category of tasks. Designers typically focus on physical field distributions and aerodynamic coefficients near the airfoil wall. When addressing active flow control or noise interference, physical parameter distributions in the wake and far-field regions become equally essential. Unlike prior studies, this research utilizes both the near-wall flow dataset and the full-field dataset made public by Bonnet et al. (2022). The full-field dataset extends beyond the near-wall data by challenging the ability of operators to model long-range interactions and large computational domains. Differences between the full-field and near-wall data are illustrated in Figure 2.

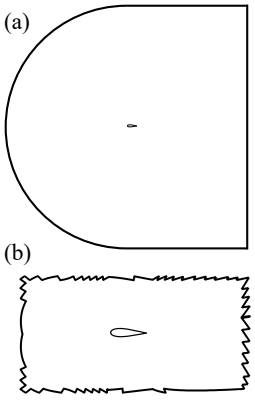

Figure 2: The difference between full-field data and near-wall data, (a) full-field, (b) near-wall.

Table 2: Model Performance Comparison. The C_p and C_l columns report the rel L_2 error of airfoil wall pressure distribution and lift coefficient, respectively.

Setting	Model	RMSE	Rel L_2	R^2	C_p	C_l
Full field	MLP	0.8852	1.0020	-0.0042	1.0709	1.4403
	U-Net	0.4540	0.5139	0.7358	0.4238	0.9239
	GraphSAGE	0.3602	0.4078	0.8336	0.3939	0.7615
	PointNet	0.5582	0.6319	0.6006	0.6129	0.9464
	FNO	0.8839	1.0007	-0.0014	1.0410	1.3927
	Transolver	0.3102	0.3511	0.8766	0.1904	0.7171
	GDDNO (ours)	**0.0720**	**0.0815**	**0.9933**	**0.0480**	**0.0735**
Near wall	MLP	0.1365	0.1367	0.9812	0.1856	0.1434
	U-Net	0.2028	0.2032	0.9586	0.2483	0.1609
	GraphSAGE	0.1952	0.1956	0.9617	0.3930	0.1346
	PointNet	0.2455	0.2460	0.9394	0.2544	0.3333
	FNO	0.6233	0.6245	0.6098	0.6408	0.3688
	Transolver	0.0453	0.0489	0.9977	0.0796	0.1142
	GDDNO (ours)	**0.0419**	**0.0402**	**0.9985**	**0.0781**	**0.0302**

As shown in Table 2, GDDNO achieves an RMSE of 0.0720 and a relative L_2 error of 0.0815. Transolver is the strongest baseline on these field metrics, with an RMSE of 0.3102 and a relative L_2 error of 0.3511. These differences correspond to reductions of 76.8% for both field metrics. GDDNO also reached R^2 to 0.9933. The relative error of C_p is reduced by 74.8%, and the C_l error is reduced by 89.8%. The improvement of GDDNO on the near-wall task is not as obvious as that of full-field task, but it still makes progress. The prediction of lift coefficient is remarkable, which especially shows that GDDNO has a strong ability to express the local details of the physical field. On the near-wall task, GDDNO achieves a relative L_2 error of 0.0402 and an R^2 of 0.9985. The combined results show that GDDNO performs strongly on the commonly used near-wall subset and provides a better choice when the far-field computational domain cannot be ignored.

4.1.2 SIMULATION-TO-REAL PREDICTION ON REALPDEBENCH

Table 3 reports the three main field metrics for numerical-data training and subsequent real-world fine-tuning. GDDNO achieves the lowest simulation-trained relative L_2 error on all five systems. Its relative L_2 errors are 0.1493 on Cylinder, 0.1036 on Controlled Cylinder, 0.1413 on FSI, 0.0344 on Foil, and 0.7314 on Combustion. The corresponding RMSE and R^2 are also the best in every simulation-trained task.

Table 3: RealPDEBench performance after simulation training and real-world fine-tuning.

Dataset	Model	Simulated training			Real-world finetuning		
		RMSE	Rel L_2	R^2	RMSE	Rel L_2	R^2
Cylinder	U-Net	0.0777	0.2257	0.6101	0.0638	**0.0754**	0.7370
	CNO	0.0660	0.1808	0.7186	0.0374	0.0849	0.9099
	DeepONet	0.0883	0.3550	0.4965	0.0677	0.1491	0.7181
	FNO	0.0736	0.2628	0.6505	0.0538	0.0770	0.8132
	WDNO	0.0699	0.2053	0.6847	0.0513	0.0969	0.8300
	MWT	0.0733	0.2240	0.7132	0.0528	0.0816	0.8202
	GK-Transformer	0.0810	0.3044	0.5765	0.0637	0.1908	0.7377
	Transolver	0.1032	0.3117	0.3128	0.0772	0.3620	0.6153
	GDDNO (ours)	**0.0426**	**0.1493**	**0.8830**	**0.0355**	0.0820	**0.9186**
Controlled Cylinder	U-Net	0.0203	0.1412	0.9187	**0.0080**	0.0559	0.9873
	CNO	0.0165	0.1202	0.9458	0.0082	0.0582	0.9869
	DeepONet	0.0523	0.3986	0.4591	0.0279	0.2214	0.8454
	FNO	0.0279	0.1992	0.8453	0.0094	0.0702	0.9824
	WDNO	0.0240	0.1829	0.8854	0.0101	0.0752	0.9799
	MWT	0.0229	0.1632	0.8962	0.0102	0.0749	0.9793
	GK-Transformer	0.0259	0.1808	0.8668	0.0102	0.0753	0.9795
	Transolver	0.0254	0.1757	0.8723	0.0166	0.1164	0.9454
	GDDNO (ours)	**0.0148**	**0.1036**	**0.9565**	**0.0080**	**0.0556**	**0.9881**
FSI	U-Net	0.0219	0.1553	0.9188	0.0085	0.0590	0.9877
	CNO	0.0245	0.1761	0.8975	0.0089	0.0633	0.9864
	DeepONet	0.0644	0.4664	0.2946	0.0330	0.2355	0.8147
	FNO	0.0432	0.3138	0.6829	0.0127	0.0881	0.9727
	WDNO	0.0369	0.2700	0.7678	0.0116	0.0824	0.9770
	MWT	0.0342	0.2503	0.8009	0.0129	0.0918	0.9717
	GK-Transformer	0.0314	0.2285	0.8325	0.0126	0.0903	0.9732
	Transolver	0.0301	0.2058	0.8457	0.0246	0.1632	0.8969
	GDDNO (ours)	**0.0199**	**0.1413**	**0.9325**	**0.0082**	**0.0576**	**0.9885**
Foil	U-Net	0.0270	0.0732	0.9323	**0.0098**	0.0156	0.9911
	CNO	0.0224	0.0436	0.9533	0.0141	0.0293	0.9815
	DeepONet	0.0345	0.0537	0.8895	0.0269	0.0516	0.9326
	FNO	0.0271	0.0544	0.9317	0.0126	0.0222	0.9863
	WDNO	0.0242	0.0490	0.9457	0.0106	0.0181	0.9895
	MWT	0.0265	0.0413	0.9347	0.0126	0.0217	0.9852
	GK-Transformer	0.0277	0.0472	0.9290	0.0134	0.0231	0.9834
	Transolver	0.0312	0.0512	0.9099	0.0197	0.0349	0.9638
	GDDNO (ours)	**0.0209**	**0.0360**	**0.9579**	**0.0098**	**0.0149**	**0.9912**
Combustion	U-Net	0.0378	0.7656	0.2056	**0.0211**	**0.5404**	**0.7520**
	CNO	0.0427	0.9182	-0.0104	0.0241	0.5885	0.6785
	DeepONet	0.0403	0.8859	0.1007	0.0231	0.5723	0.7033
	FNO	0.0367	0.7915	0.2529	0.0224	0.5608	0.7220
	WDNO	0.0439	1.1016	-0.0695	0.0314	0.7441	0.4533
	MWT	0.0387	0.7848	0.1679	0.0217	0.5507	0.7380
	GK-Transformer	0.0401	0.8455	0.1082	0.0246	0.6109	0.6657
	Transolver	0.0442	0.9567	-0.0818	0.0382	0.7891	0.1914
	GDDNO (ours)	**0.0359**	**0.7314**	**0.2852**	0.0216	0.5449	0.7422

Real-world fine-tuning reduces the GDDNO's RMSE on every system. The largest reductions were observed on FSI and Foil, where RMSE decreased from 0.0199 to 0.0082 and from 0.0209 to 0.0098. The lowest fine-tuned relative L_2 errors are obtained by GDDNO on Controlled Cylinder, FSI, and Foil. U-Net retained a lower relative L_2 error on Cylinder and Combustion. GDDNO still produced the highest fine-tuned R^2 on Cylinder, Controlled Cylinder, and FSI. This pattern indicates that local subdomain modeling and compressed global communication remained effective across flow, fluid-structure, and combustion dynamics. RealPDEBench is also an autoregressive task. Autoregressive evaluation on RealPDEBench assesses temporal prediction and transfer from simulated to experimental observations

4.1.3 CAVITY COMBUSTION FLOW

he cavity combustion dataset tests coupled compressible flow and chemical reactions on three-dimensional unstructured meshes. Table 4. reports the results. GDDNO produces the lowest relative L_2 error of 0.4148. Its R^2 of 0.6612 exceeds the strongest baseline value of 0.6483 reported by GraphSAGE. The combustion efficiency error is 0.2112. The combustion-efficiency error provides an additional assessment of the predicted reacting flow fields, which demonstrates the excellent performance of GDDNO in complex multi-physics coupling scenarios.

Table 4: Model performance on reacting flow. The η is the relative L_2 error of combustion efficienc.

Model	RMSE	Rel L_2	R^2	η
U-Net (Graph, 3D)	0.3812	0.4630	0.5885	0.2348
GraphSAGE	0.3488	0.4237	0.6483	0.2328
PointNet	0.3882	0.4699	0.5832	0.3487
FNO	0.4047	0.4856	0.5609	0.2307
Transolver	0.5381	0.6286	0.4217	0.2495
GDDNO (ours)	**0.3413**	**0.4148**	**0.6612**	**0.2112**

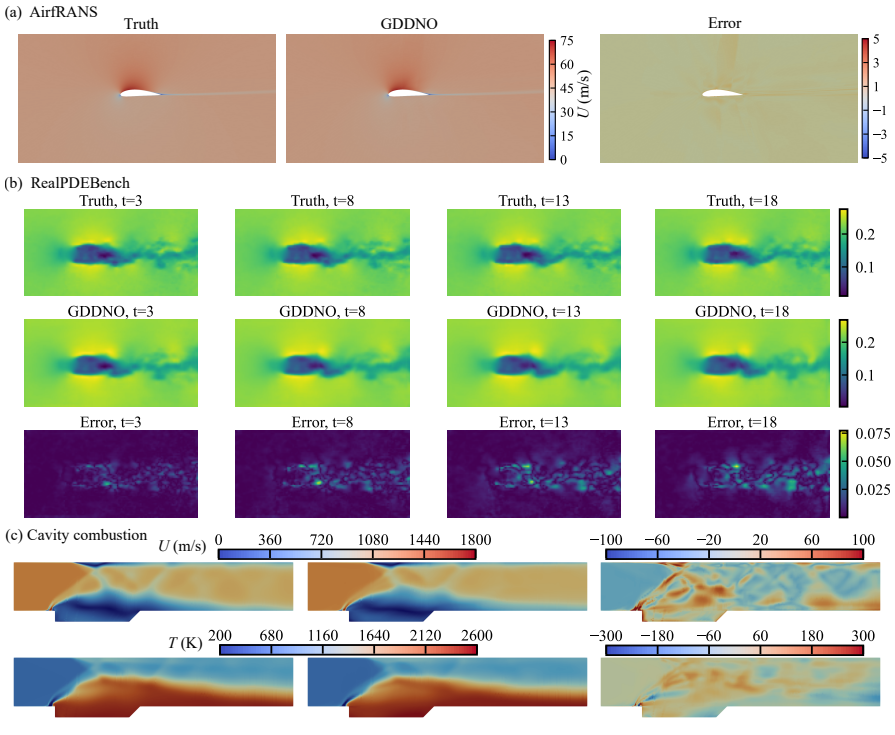

Figure 3: Representative predictions and errors for GDDNO on (a) AirfRANS, (b) RealPDEBench, and (c) cavity combustion. Additional visualizations are provided in Appendix E.

4.2 MODEL ABLATIONS

Ablation experiments on AirfRANS examine the effects of local modeling and global assembly on prediction accuracy. Additional ablations of subdomain count and overlap depth are reported in Appendix C.

As shown in Table 5, removing edge attributes increase RMSE by 12.8% and relative L_2 error by 12.8%. Replacing the full edge operator with node encode increase RMSE by 18.3%, C_p error by 34.5%, and C_l error by 14.1%. The larger degradation indicates that mesh topology and the edge-conditioned message function jointly contribute to local flow modeling.

Removing overlapping cells and the interface consistency loss increase field RMSE and relative L_2 error by 12.0%. The effect is larger for aerodynamic coefficients. The C_p error increased by 46.4%, and the C_l error increased by 50.3%. These outputs depend strongly on surface and near-surface structure. The results support the important role of interface flux consistency constraint in preventing the influence of artificial subdomain boundaries.

Table 5: Ablation study of different GDDNO variants.

Variant	RMSE	Rel L_2	R^2	C_p	C_l
GDDNO	**0.0721**	**0.0816**	**0.9934**	**0.0480**	**0.0736**
Remove mesh topology	0.0813	0.0920	0.9915	0.0486	0.0816
Only node encode	0.0853	0.0945	0.9865	0.0646	0.0840
Remove overlapping cells and interface consistency loss	0.0807	0.0914	0.9917	0.0703	0.1106

5 CONCLUSION

This paper proposes GDDNO, a neural operator that adopts domain decomposition to combine local geometric detail with global coupling. Cell-level features are retained while global interactions are learned through compact subdomain tokens. Shared-cell assembly and consistency losses coordinate local predictions across artificial boundaries. Improvements were observed in field predictions across the tested systems and in engineering quantities. GDDNO achieves the lowest relative L_2 error among the compared models on all five RealPDEBench systems after training on simulated data. On full-field AirfRANS, the relative L_2 error is reduced from 0.3511 for the strongest baseline to 0.0815 and achieves the best aerodynamic coefficients prediction. The lowest field and combustion-efficiency relative L_2 errors are also obtained on the three-dimensional cavity combustion flow. GDDNO could support the development of surrogate models for repeated design evaluations involving complex geometries and coupled physical processes. Furthermore, this work is inspired by traditional numerical simulation methods, which led us to reexamine the strategies for modeling PDE operators. Numerical frameworks such as the finite-volume method and finite-element analysis provide both computational strategies and architectural ideas for scientific machine learning.

AI USE STATEMENT

In this work, we used generative AI tools for [tasks with required disclosure]. We have not used generative AI tools for [other tasks with required disclosure], and [the rest of the required disclosure tasks] are not applicable to this work. Additionally, we used generative AI tools for [tasks with recommended disclosure]. We have reviewed all AI-assisted work. [Elaborate. For example, "we checked LLM-generated research ideas for potential plagiarism through a manual literature survey", "LLM-generated code was verified and tested for correctness by 2 authors", etc.]. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we provide comprehensive experimental specifications in the paper and supplementary materials. All detailed training protocols, ablation experiment re-

sults and evaluation metric descriptions are documented in Appendix B and C. The AirfRANS and PealPDEBench datasets used in this paper are based on publicly available classic datasets from previous studies with standardized data processing pipelines (Bonnet et al., 2022; Hu et al., 2026a). The cavity combustion dataset was generated by extending previous work (Zhao et al., 2026; Wang et al., 2013b). Appendix B introduces the source of the dataset in detail. The source code of this paper will be publicly released within two weeks after the paper is accepted, and the repository link will be added to the final camera-ready version.

REFERENCES

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 25152–25194. Curran Associates, Inc., 2024. doi: 10.52202/079017-0793.

Florent Bonnet, Jocelyn Mazari, Paola Cinnella, and Patrick Gallinari. AirfRANS: High Fidelity Computational Fluid Dynamics Dataset for Approximating Reynolds-Averaged Navier– Stokes Solutions. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23463–23478, New Orleans, 2022. Curran Associates, Inc.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message Passing Neural PDE Solvers. In *International Conference on Learning Representations*, October 2021.

Shuhao Cao. Choose a transformer: Fourier or galerkin. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 24924–24940. Curran Associates, Inc., 2021.

Tony F. Chan and Tarek P. Mathew. Domain decomposition algorithms. *Acta Numerica*, 3:61–143, January 1994. ISSN 1474-0508, 0962-4929. doi: 10.1017/S0962492900002427.

R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.16.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 24048–24062. Curran Associates, Inc., 2021.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1025–1035, Red Hook, NY, USA, December 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.

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*, pp. 12556–12569. PMLR, July 2023.

Alexander Heinlein, Axel Klawonn, Martin Lanser, and Janine Weber. Combining machine learning and domain decomposition methods for the solution of partial differential equations—A review. *GAMM-Mitteilungen*, 44(1):e202100001, 2021. ISSN 1522-2608. doi: 10.1002/gamm.202100001.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8.

Peiyan Hu, Rui Wang, Xiang Zheng, Tao Zhang, Haodong Feng, Ruiqi Feng, Long Wei, Yue Wang, Zhi-Ming Ma, and Tailin Wu. Wavelet Diffusion Neural Operator. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 12291–12333, 2025.

Peiyan Hu, Haodong Feng, Hongyuan Liu, Tongtong Yan, Wenhao Deng, Tianrun Gao, Rong Zheng, Haoren Zheng, Chenglei Yu, Chuanrui Wang, Kaiwen Li, Zhi-Ming Ma, Dezhi Zhou, Xingcai Lu, Dixia Fan, and Tailin Wu. RealPDEBench: A benchmark for complex physical systems with real-world data. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 107143–107188, 2026a.

Wenjie Hu, Sidun Liu, Peng Qiao, Zhenglun Sun, and Yong Dou. Transolver Is a Linear Transformer: Revisiting Physics-Attention Through the Lens of Linear Attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 408–416, March 2026b. doi: 10.1609/aaai.v40i1.37003.

Jianing Huang, Kaixuan Zhang, Youjia Wu, and Ze Cheng. Operator learning with domain decomposition for geometry generalization in PDE solving. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 69121–69148, 2026.

Ameya D. Jagtap and George Em Karniadakis. Extended Physics-Informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition Based Deep Learning Framework for Nonlinear Partial Differential Equations. *Communications in Computational Physics*, 28(5): 2002–2041, November 2020. ISSN 1815-2406, 1991-7120. doi: 10.4208/cicp.OA-2020-0164.

Ameya D. Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, June 2020. ISSN 0045-7825. doi: 10.1016/j.cma.2020.113028.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, June 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Jielu Li, Mingming Guo, and Ye Tian. Recent application progress of graph neural networks in computational fluid dynamics. *Aerospace Science and Technology*, 179:113621, December 2026. ISSN 1270-9638. doi: 10.1016/j.ast.2026.113621.

Wuyang Li, Xueshuang Xiang, and Yingxiang Xu. Deep Domain Decomposition Method: Elliptic Problems. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, pp. 269–286. PMLR, August 2020a.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for Partial Differential Equations' Operator Learning. *Transactions on Machine Learning Research*, December 2022. ISSN 2835-8856.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole Graph Neural Operator for Parametric Partial Differential Equations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6755–6766. Curran Associates, Inc., 2020b.

Zongyi Li, Nikola Borislavov 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*, January 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(1):388:18593–388:18618, January 2023a. ISSN 1532-4435.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-Informed Neural Operator for Large-Scale 3D PDEs. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 35836–35854. Curran Associates, Inc., 2023b.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS J. Data Sci.*, 1(3), May 2024. doi: 10.1145/3648506.

Songming Liu, Zhongkai Hao, Chengyang Ying, Hang Su, Ze Cheng, and Jun Zhu. NUNO: A General Framework for Learning Parametric PDEs with Non-Uniform Data. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 21658–21671. PMLR, July 2023.

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, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5.

Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (FBPINNs): A scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, 49(4):62, August 2023. ISSN 1019-7168, 1572-9044. doi: 10.1007/s10444-023-10065-9.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning Mesh-Based Simulation with Graph Networks. In *International Conference on Learning Representations*, October 2020.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. ISSN 00219991. doi: 10.1016/j.jcp.2018.10.045.

Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of pdes. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 77187–77200. Curran Associates, Inc., 2023. doi: 10.52202/075280-3376.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28.

Andrea Toselli and Olof B. Widlund. *Domain Decomposition Methods — Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 2005. ISBN 978-3-540-20696-5 978-3-540-26662-4. doi: 10.1007/b137868.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Hongbo Wang, Zhenguo Wang, Mingbo Sun, and Ning Qin. Combustion characteristics in a supersonic combustor with hydrogen injection upstream of cavity flameholder. *Proceedings of the Combustion Institute*, 34(2):2073–2082, January 2013a. ISSN 1540-7489. doi: 10.1016/j.proci.2012.06.049.

Hongbo Wang, Zhenguo Wang, Mingbo Sun, and Haiyan Wu. Combustion modes of hydrogen jet combustion in a cavity-based supersonic combustor. *International Journal of Hydrogen Energy*, 38(27):12078–12089, September 2013b. ISSN 03603199. doi: 10.1016/j.ijhydene.2013.06.132.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, Vienna, Austria, 2024. JMLR.org.

Tong Zhao, Jiajian Li, Yvpeng Liu, Boxian Yin, Bing Liu, Shaohua Zhu, Jian An, and Junqiang Bai. Physics latent manifold-guided flow fields reconstruction for accelerating supersonic combustion simulations. *Aerospace Science and Technology*, 179:113391, December 2026. ISSN 12709638. doi: 10.1016/j.ast.2026.113391.

Kuijun Zuo, Zhengyin Ye, Weiwei Zhang, Xianxu Yuan, and Linyang Zhu. Fast aerodynamics prediction of laminar airfoils based on deep attention network. *Physics of Fluids*, 35(3):037127, March 2023. ISSN 1070-6631. doi: 10.1063/5.0140545.

A ADDITIONAL DETAILS OF GDDNO

A.1 FROM MESH DECOMPOSITION TO LOCAL COMPUTATION

Let B_p denote the core cells of subdomain p, and let E be the directed edges of the global mesh graph. The cores are disjoint and cover the mesh. With overlap depth H, the expanded cell and edge sets are

$$V_p = \{i : \mathrm{dist}_{\mathrm{mesh}}(i, B_p) \le H\}, \qquad E_p = \{(j, i) \in E : j \in V_p,\ i \in V_p\}.$$

The distance counts the fewest adjacency hops needed to reach the core. Neighboring cells are added to create overlap between subdomains. Every physical cell is included at least once. Copies in different subdomains refer to the same cell in the original mesh and are matched by its global identifier.

The neighborhood in Section 3.3 is $\mathcal{N}_p(i) = \{j : (j, i) \in E_p\}$. One layer receives information from immediate neighbors. Each additional layer extends this range by one hop, so after ℓ layers, the feature at cell i depends only on inputs within ℓ hops in the subdomain graph. Halo cells extend this range beyond the core boundary, but all paths must remain within V_p.

The same cell may receive different information in different subdomains because its available neighbors differ. Its features and predictions can therefore differ even when the network parameters are shared. The overlap consistency objective in Section 3.5 penalizes disagreement between these predictions.

A.2 FROM POOLED TOKENS TO GLOBAL CONTEXT

Each subdomain is first represented by K tokens. For each pooling query, the weights in Section 3.4 satisfy

$$\alpha_{p,k,i} \ge 0, \qquad \sum_{i \in V_p} \alpha_{p,k,i} = 1.$$

Each token is a weighted average of projected cell features, followed by an output projection. Different queries can assign different weights to the same cells, allowing the tokens to summarize different parts of a subdomain. The original cell features are retained for the final prediction.

The tokens exchange information through attention. For one head in one layer, let $v_{q,l}$ be the value vector obtained by normalizing and projecting token l of subdomain q. The output for token k of subdomain p is $\sum_{q,l} A_{p,k;q,l} v_{q,l}$. Averaging over the receiving tokens in subdomain p gives

$$\beta_{p;q,l} = \frac{1}{K} \sum_{k=1}^{K} A_{p,k;q,l}, \qquad \overline{z}_p^{\mathrm{attn}} = \sum_{q=1}^{P} \sum_{l=1}^{K} \beta_{p;q,l} v_{q,l}, \qquad \sum_{q,l} \beta_{p;q,l} = 1.$$

The sums are rearranged by collecting the coefficient of each source value $v_{q,l}$ over the K receiving tokens. This coefficient, $\beta_{p;q,l}$, is the average attention assigned to that source. Information from subdomain q is pooled into tokens and passed to subdomain p through these weights. The final tokens of subdomain p are averaged to obtain $\overline{z}_p$. Each cell is decoded from this common context and its own feature $h_{p,i}$. The context supplies global information, while the cell features retain local spatial differences.

A.3 ASSEMBLY AND THE CONSISTENCY OBJECTIVES

Shared-cell assembly. The predictions for the same physical cell are matched by global identifier. If a cell belongs to only one subdomain, its prediction is unchanged. If a cell belongs to multiple subdomains, its predictions are averaged as described in Section 3.5. Even if the prediction is inconsistent, the overlap loss solves the difference.

Data and consistency losses. The data, overlap, and interface losses are:

$$\mathcal{L}_{\text{data}} = \frac{1}{N}\sum_{i=1}^{N} \text{MSE}(\widehat{u}_i, u_i)$$
$$\mathcal{L}_{\text{overlap}} = \text{mean}_{(p,i):\, i\in V_p,\, m_i>1}\, \text{MSE}(\widehat{u}_{p,i}, \widehat{u}_i)$$
$$\mathcal{L}_{\text{interface}} = \text{mean}_{(i,j)\in\mathcal{I}}\, \text{MSE}(\widehat{u}_i, \widehat{u}_j)$$

where $\mathcal{I}$ contains neighboring cell pairs across core boundaries. The data loss is averaged over physical cells. The overlap loss compares each shared-cell prediction with its assembled mean. The interface loss compares distinct neighboring cells across a core boundary.

The role of the overlap loss can be seen by writing $\widehat{u}_{p,i} - u_i = (\widehat{u}_{p,i} - \widehat{u}_i) + (\widehat{u}_i - u_i)$. Each local error is the sum of its deviation from the assembled prediction and the error of that prediction. Squaring and averaging over copies of cell i gives:

$$\frac{1}{m_i}\sum_{p\in\mathcal{C}(i)} \|\widehat{u}_{p,i} - u_i\|_2^2 = \|\widehat{u}_i - u_i\|_2^2 + \frac{1}{m_i}\sum_{p\in\mathcal{C}(i)} \|\widehat{u}_{p,i} - \widehat{u}_i\|_2^2$$

The cross terms vanish because the deviations from the average sum to zero, $\sum_{p\in\mathcal{C}(i)}(\widehat{u}_{p,i}-\widehat{u}_i) = 0$. The first term on the right measures the error of the assembled prediction, and the second measures disagreement among its local copies. The data loss constrains the first quantity and the overlap loss the second. Together, the two losses penalize both an inaccurate common prediction and inconsistent predictions whose errors cancel during assembly.

The interface term acts on neighboring states used in the finite-volume flux calculation. Penalizing differences across the artificial boundary supports the flux-continuity goal in Section 3.2 through state consistency. Neighboring cells can have different physical states, so the interface weight could be chosen with this variation in mind. Both consistency weights can be adjusted for the problem, including setting a weight to zero.

B Implementation details

B.1 Bechmarks

Three different benchmarks are used to evaluate GDDNO. AirfRANS emphasizes geometry-sensitive steady prediction and near-wall force recovery. RealPDEBench emphasizes autoregressive stability and transfer from numerical to experimental distributions. Cavity combustion emphasizes large irregular meshes and multiphysics coupling.

AirfRANS. AirfRANS was introduced by Bonnet et al. (2022) as a high-fidelity dataset for learning steady solutions of the two-dimensional incompressible Reynolds-averaged Navier-Stokes equations around airfoils. The flow is simulated under sea-level conditions at a temperature of 298.15 K. Reynolds numbers range from 2×10^6 to 6×10^6, and the corresponding Mach number remains below 0.3 so that the incompressible approximation remains appropriate. Angles of attack range from -5° to 15°. Airfoil geometries are sampled from the parameterized NACA four-digit and five-digit families. A four-digit airfoil is represented by $MPXX$, where M specifies the maximum camber in hundredths of the chord, P specifies the position of maximum camber in tenths of the chord, and XX specifies maximum thickness in hundredths of the chord. A five-digit airfoil is represented by $LPQXX$, where L and P determine the camber-line construction, $Q \in \{0, 1\}$ distinguishes single and double-cambered profiles, and XX again specifies thickness.

The meshes are generated with `blockMesh` in OpenFOAM v2112. A multiblock C-grid is constructed for each airfoil following the topology used in NASA reference cases. Each original mesh contains approximately 250,000 to 300,000 cells. Steady solutions are obtained using `simpleFOAM` with the k-ω SST turbulence model.

The official AirfRANS provides 1000 samples and defines multiple partitions, we used three of them. In the main text, we report the full task, and the other two tasks are reported in Appendix D.1:

- Full task: The full data task uses 800 samples for training and 200 for testing.
- Aoa task: For angle of attack extrapolation, models are trained on $\alpha \in [-2.5^\circ, 12.5^\circ]$ and tested on $\alpha \in [-5^\circ, -2.5^\circ]$ and $\alpha \in [12.5^\circ, 15^\circ]$.
- Reynolds task: For Reynolds number extrapolation , models are trained on $Re \in [3 \times 10^6, 5 \times 10^6]$ and tested on the two outer intervals $Re \in [2 \times 10^6, 3 \times 10^6]$ and $Re \in [5 \times 10^6, 6 \times 10^6]$.

RealPDEBench. RealPDEBench was introduced by Hu et al. (2026a) to evaluate scientific machine-learning models with paired real-world measurements and numerical simulations. The benchmark contains five physical scenarios, 368 experimental trajectories, and 368 paired simulated trajectories. Each trajectory contains more than 2,000 frames. The total of 736 trajectories permits evaluation of prediction from real data, direct transfer from simulation to reality, and simulated pretraining followed by real-data fine-tuning.

RealPDEBench contains five scenarios: cylinder, controlled cylinder, fluid-structure interaction (FSI), foil, and combustion. Simulated data contain modalities that are unavailable in experiments. During simulated training, additional modalities are randomly masked. At real-data evaluation, unavailable channels are replaced by zeros. Noise can be added to simulated trajectories to reduce the distribution mismatch caused by clean numerical fields and noisy measurements. The framework supports no normalization, Gaussian standardization, and normalization by maximum absolute value. All compared models in the present study used the officially provided pipeline for normalizing statistics and channel masking.

Cavity combustion flow. The cavity combustion flow benchmark is derived from the hydrogen-fueled supersonic combustor studied by Wang et al. (2013b) and Zhao et al. (2026). A wall-mounted cavity promotes fuel-air mixing and flame stabilization under a short residence time. Hydrogen is injected upstream of the cavity. Four geometric parameters are varied, including the streamwise distance between the injection location and cavity, cavity depth, cavity length, and cavity trailing-edge angle. The air inlet has a total temperature of $T_0 = 1486$ K, a total pressure of $p_0 = 1.6$ MPa, and a Mach number of $Ma = 2.52$. The hydrogen inlet has $T_0 = 300$ K, $p_0 = 1.2$ MPa, $Ma = 1.0$, and a hydrogen mass fraction of 100%.

The numerical setup followed Zhao et al. (2026) and Wang et al. (2013a), and additional simulations extend the existing dataset. The simulations are performed in OpenFOAM with a second-order total-variation-diminishing scheme and a maximum Courant number of 0.6. Reynolds-averaged Navier-Stokes equations are coupled to the partially stirred reactor combustion model and the k-ω SST turbulence model. Hydrogen chemistry follows the mechanism adopted by Zhao et al. The numerical setup was validated against measured streamwise wall-pressure distributions before the dataset was generated. Each source-paper configuration contains approximately 600,000 finite-volume cells. The spanwise width is 30 mm, and periodic spanwise boundaries are used to reduce computational cost while preserving the reported flame distribution. A total of 826 samples spanning different cavity geometries were generated, with 726 used for training and evaluation, and 90 for testing.

B.2 METRICS

Our experiments cover three distinct physical systems. Relative L_2 error, RMSE, and R^2 are used to measure field accuracy across the systems, while additional physical metrics are used to evaluate quantities specific to each system.

B.2.1 GENERAL FIELD METRICS

Relative L_2 error. This dimensionless metric measures the prediction error relative to the magnitude of the reference field. Lower values indicate better agreement.

$$\mathrm{Rel}\, L_2 = \frac{\|\hat{y} - y\|_2}{\|y\|_2}$$

Root mean squared error (RMSE). RMSE measures the typical magnitude of the error in the units of the evaluated field. Squaring gives larger errors greater influence, and lower values indicate better

predictions.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

Coefficient of determination (R^2). R^2 measures how much of the variation in the reference data is captured by the prediction. A value of 1 indicates an exact prediction, 0 corresponds to using the reference mean, and a negative value indicates performance worse than that baseline. Higher values are preferred.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

B.2.2 AIRFRANS METRICS

Wall pressure-coefficient. The pressure coefficient is evaluated along the airfoil wall. For the kinematic pressure p_k used in AirfRANS, it is calculated as:

$$C_p = \frac{2(p_k - p_{k,\infty})}{U_\infty^2}$$

where U_∞ is the freestream speed and $p_{k,\infty}$ is the reference kinematic pressure. The relative L_2 error between the predicted and reference C_p distributions measures how accurately the model reproduces the local surface-pressure pattern. This distribution is physically informative because the pressure difference between the upper and lower airfoil surfaces contributes strongly to lift.

Lift-coefficient. The lift coefficient is obtained from the integrated wall force, including pressure and viscous contributions:

$$C_l = \frac{\boldsymbol{F}\cdot\boldsymbol{e}_l}{q_\infty A_{\mathrm{ref}}}, \qquad q_\infty = \frac{1}{2}\rho_\infty U_\infty^2$$

where $\boldsymbol{e}_l$ is the lift-direction unit vector and A_{ref} is the reference area. The reported error is the relative L_2 error between predicted and reference C_l values across cases. It assesses whether field predictions preserve the integrated aerodynamic force, rather than only matching local flow values.

B.2.3 REALPDEBENCH METRICS

These metrics proposed by the official(Hu et al., 2026a).

Kinetic energy error (KE). For velocity components u and v, temporal fluctuations are defined at each spatial location as $u' = u - \bar{u}$ and $v' = v - \bar{v}$, where the overbar denotes a time average. The fluctuation kinetic energy is:

$$e = \frac{1}{2}\left(\overline{u'^2} + \overline{v'^2}\right), \qquad \mathrm{KE} = |e - \hat{e}|$$

Frequency error (FE). The predicted and reference fields are first summed over spatial locations to form temporal signals. A one-dimensional Fourier transform is then applied, and the mean absolute difference between the resulting spectra is computed:

$$\mathrm{FE} = \frac{1}{KT}\sum_{k=1}^{K}\sum_{t=1}^{T}\left|\mathcal{F}_t\left(\sum_i y_k(t,x_i)\right) - \mathcal{F}_t\left(\sum_i \hat{y}_k(t,x_i)\right)\right|$$

Here K is the number of evaluated samples and T is the number of temporal entries.

Fourier-space RMSE (fRMSE). Three-dimensional fast Fourier transforms are applied to the prediction and reference fields. Fourier coefficients are grouped by frequency magnitude, and the root

mean squared coefficient difference is computed within each frequency group B_r:

$$\mathrm{fRMSE}(r) = \sqrt{\frac{1}{|B_r|} \sum_{\boldsymbol{q} \in B_r} \left| \widehat{Y}(\boldsymbol{q}) - Y(\boldsymbol{q}) \right|^2}$$

Here Y and $\widehat{Y}$ are the Fourier transforms of the reference and predicted fields, and r denotes a frequency-magnitude group.

B.2.4 CAVITY-COMBUSTION METRIC

Combustion Efficiency. The combustion efficiency based on chemical enthalpy flux is evaluated at a downstream section x as:

$$\eta_{c,H}(x) = \frac{\dot{H}_{c,\mathrm{in}} - \dot{H}_c(x)}{q_F \dot{m}_F}, \qquad h_c = \sum_s \Delta h_s^\theta y_s, \qquad \dot{H}_c(x) = \int_x h_c \rho u_n \, \mathrm{d}A$$

where h_c is the mixture chemical enthalpy, y_s is the mass fraction of species s, Δh_s^θ is its standard formation enthalpy, q_F is the fuel heating value, and $\dot{m}_F$ is the fuel mass-flow rate. The efficiency estimates the fraction of the fuel's reference chemical power released between the inlet and section x. The experiment reports the relative L_2 error between the predicted and reference efficiencies for each case, averaged across cases. Lower error indicates more accurate prediction of this integral combustion outcome.

C ADDITIONAL ABLATIONS

The additional experiments examine how prediction accuracy depends on subdomain count and overlap depth. These choices determine the spatial region encoded by each local geometry-aware operator and the shared support available for interface coordination.

Number of subdomains. The lowest field errors in the subdomain-count ablation are obtained at $P = 32$. Six decompositions were evaluated, with $P \in \{8, 16, 24, 32, 40, 48\}$ and two layers overlap. As shown in Table 6 and Figure 4, the lowest RMSE and relative L_2 error were 0.0720 and 0.0816, respectively, and the highest R^2 was 0.9933. Relative to $P = 8$, RMSE was reduced by 0.0244. Relative to $P = 16$, both RMSE and relative L_2 error were reduced by 0.0132, and the C_p error was reduced by 0.0084.

Table 6: Performance comparison across different subdomain counts

Subdomains N_P	RMSE	Rel L_2	R^2	C_p	C_l
8	0.0964	0.1091	0.9881	0.0512	0.0678
16	0.0852	0.0965	0.9907	0.0564	0.0669
24	0.0923	0.1044	0.9891	0.0496	0.0820
32	**0.0720**	**0.0816**	**0.9933**	0.0480	0.0735
40	0.0889	0.1006	0.9898	**0.0475**	**0.0687**
48	0.0863	0.0976	0.9788	0.0483	0.0871

Accuracy does not improve monotonically as the mesh is divided into more subdomains. Increasing N_P from 16 to 32 increased the accuracy, but the gain is not retained at N_P=40 or 48. Ideally, prediction accuracy should be insensitive to the number of subdomains. Even if the deviation between the current number of subdomains is at a relatively small level, the current experiment still shows the difference. The decomposition changes both the spatial extent of local computation and the organization of the inputs to the global communicator. At a fixed number of tokens per subdomain, it also changes the total token count. Improving robustness to decomposition choices remains a direction for our future work.

Overlap depth. Overlap depths $H \in \{0, 1, 2, 3, 4\}$ were evaluated at $P = 16$, 32, and 48. Table 7 and Figure 5 report all results for these combinations. The overlap provides neighboring cells

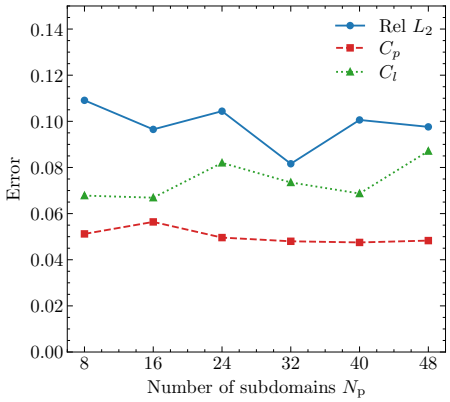

Figure 4: Effect of the number of subdomains.

for local propagation and shared states for interface flux coordination. Prediction accuracy does not improve monotonically with overlap depth. When the overlap depth is large, it also have an adverse effect on the calculation accuracy. This is mainly because the purpose of overlapping assembly is to eliminate the artificial interface between subdomains, but the larger depth will make the model ignore the physical characteristics of the subdomain core. Therefore, overlap depth should be considered together with decomposition granularity and the target physical quantities.

Table 7: Ablation Study on Subdomains P and Halo Depth H.

Subdomains N_P	Overlap depth H	RMSE	Rel L_2	R^2	C_p	C_l
16	0	0.0911	0.1031	0.9894	0.0766	0.1397
	1	0.0854	0.0967	0.9907	0.0540	0.0690
	2	0.0852	0.0965	0.9907	0.0564	0.0669
	3	0.0899	0.1018	0.9896	0.0579	0.0948
	4	0.0917	0.1038	0.9892	0.0581	0.0975
32	0	0.0807	0.0914	0.9917	0.0703	0.1106
	1	0.0747	0.0846	0.9928	0.0531	0.0895
	2	0.0720	0.0816	0.9933	0.0480	0.0735
	3	0.0705	0.0798	0.9936	0.0478	0.0721
	4	0.0760	0.0806	0.9926	0.0521	0.0796
48	0	0.0924	0.1146	0.9722	0.0593	0.0742
	1	0.0997	0.1059	0.9873	0.0510	0.0651
	2	0.0863	0.0976	0.9905	0.0483	0.0679
	3	0.0862	0.0976	0.9905	0.0480	0.0497
	4	0.0846	0.0958	0.9908	0.0489	0.0556

D Additional Experiments

D.1 Out-of-distribution (OOD) generalization

Generalization beyond the training range was examined through two AirfRANS tasks defined by angle of attack and Reynolds number. In each task, training was restricted to an interior parameter interval, and evaluation was assigned to the two outer intervals. These settings test the learned dependence on operating conditions beyond the range represented during training.

For angle-of-attack extrapolation, the training and test ranges were:

$$\alpha_{\text{train}} \in [-2.5^\circ, 12.5^\circ], \qquad \alpha_{\text{test}} \in [-5^\circ, -2.5^\circ] \cup [12.5^\circ, 15^\circ]$$

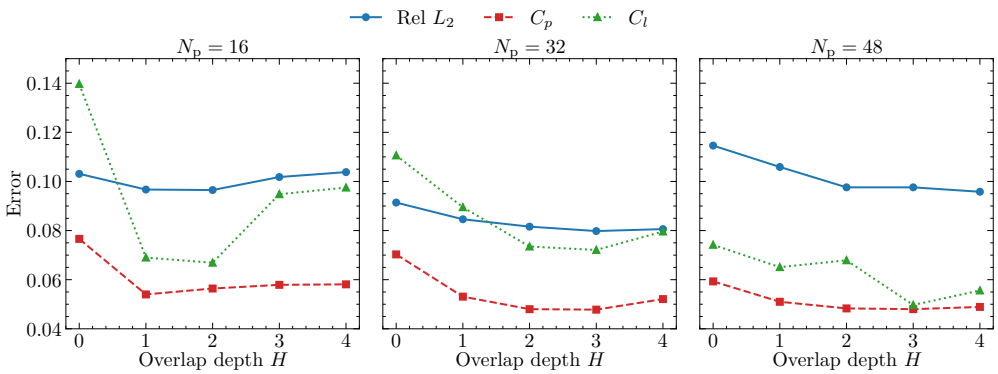

Figure 5: Effect of subdomain count and overlap depth.

For Reynolds-number extrapolation, the prescribed ranges are

$$Re_{\text{train}} \in [3 \times 10^6, 5 \times 10^6], \qquad Re_{\text{test}} \in [2 \times 10^6, 3 \times 10^6] \cup [5 \times 10^6, 6 \times 10^6]$$

The results are shown in Table 8 and 9. GDDNO achieves the best performance among the compared models on all five metrics in both extrapolation tasks. This advantage is consistent with the geometry-aware representation, through which local flow responses can be learned jointly from mesh geometry and boundary-condition inputs. Edge features retain the spatial relationships between neighboring cells, while global token communication carries information across subdomains, allowing local predictions to account for changes elsewhere in the flow field. Shared-cell assembly and the interface objective coordinate these predictions across artificial boundaries. These mechanisms help retain local detail and global consistency under changes in the operating conditions.

Table 8: OOD generalization on aoa task.

Model	RMSE	Rel L_2	R^2	C_p	C_l
MLP	0.9347	0.8935	0.1360	0.8318	0.9327
U-Net	0.5126	0.3505	0.8771	0.3362	0.3878
GraphSAGE	0.5153	0.4189	0.8245	0.5552	0.7667
PointNet	0.9210	0.6297	0.6033	0.6661	0.7173
FNO	1.3910	0.9509	0.0953	0.9729	0.9172
Transolver	0.6244	0.4269	0.8177	0.2425	0.3074
GDDNO	**0.3852**	**0.2633**	**0.9306**	**0.1199**	**0.1163**

Table 9: OOD generalization on reynolds task

Model	RMSE	Rel L_2	R^2	C_p	C_l
MLP	0.3989	0.3863	0.8507	0.4348	0.3991
U-Net	0.3135	0.3036	0.9078	0.3421	0.4150
GraphSAGE	0.5068	0.4908	0.7591	0.6789	0.4939
PointNet	0.6774	0.6561	0.5695	0.7415	0.8805
FNO	0.9943	0.9629	0.0728	0.9617	0.8733
Transolver	0.5931	0.5744	0.6701	1.1378	1.2728
GDDNO	**0.1618**	**0.1567**	**0.9755**	**0.1202**	**0.1297**

D.2 OTHER REALPDEBENCH METRICS

Additional RealPDEBench results for fRMSE, KE, MAE, and FE are reported are reported here.

Table 10: fRMSE, MAE, KE, and FE on RealPDEBench.

Dataset	Model	Simulated training				Real-world Finetuning			
		fRMSE	MAE	KE	FE	fRMSE	MAE	KE	FE
Cylinder	U-Net	0.01244	0.03036	**0.00040**	193.78	0.00982	0.01256	**0.00028**	137.78
	CNO	0.01025	0.02291	0.00043	144.86	0.00585	0.01059	0.00041	118.79
	DeepONet	0.01343	0.04282	0.00050	166.51	0.01092	0.01903	0.00045	130.29
	FNO	0.01149	0.03661	0.00047	259.03	0.00802	0.01069	0.00066	139.51
	WDNO	0.01050	0.02762	0.00079	191.33	0.00730	0.01396	0.00054	124.44
	MWT	0.01070	0.02454	0.00047	**105.18**	0.00803	0.01118	0.00035	**74.68**
	GK-Transformer	0.01267	0.04062	0.00052	246.22	0.00915	0.02154	0.00073	165.52
	Transolver	0.01649	0.04414	0.00049	446.37	0.01119	0.03425	0.00051	137.95
	HPNO(ours)	**0.00681**	**0.01717**	0.00045	167.22	**0.00552**	**0.00968**	0.00039	104.86
Controlled Cylinder	U-Net	0.00312	0.01085	**0.00005**	46.67	0.00098	0.00389	**0.00003**	21.80
	CNO	0.00218	0.00900	**0.00005**	**32.70**	0.00093	0.00409	**0.00003**	16.22
	DeepONet	0.01318	0.03689	0.00007	176.46	0.00491	0.01779	0.00006	40.83
	FNO	0.00568	0.01730	0.00007	188.92	0.00108	0.00514	0.00004	21.31
	WDNO	0.00370	0.01494	0.00009	77.30	0.00130	0.00547	0.00004	28.89
	MWT	0.00427	0.01380	0.00006	97.91	0.00106	0.00562	0.00004	20.20
	GK-Transformer	0.00503	0.01531	0.00006	119.27	0.00117	0.00555	0.00004	21.22
	Transolver	0.00529	0.01535	0.00006	81.17	0.00203	0.00911	0.00005	25.67
	HPNO(ours)	**0.00201**	**0.00778**	**0.00005**	41.64	**0.00092**	**0.00376**	**0.00003**	**14.78**
FSI	U-Net	**0.00249**	0.01267	**0.00008**	22.90	0.00070	0.00462	**0.00004**	5.48
	CNO	0.00279	0.01437	**0.00008**	**19.32**	0.00073	0.00522	0.00005	8.64
	DeepONet	0.00973	0.04344	0.00010	52.07	0.00475	0.02021	0.00009	22.40
	FNO	0.00594	0.02815	0.00009	31.57	0.00120	0.00744	0.00006	11.79
	WDNO	0.00500	0.02435	0.00019	45.81	0.00110	0.00697	0.00006	12.17
	MWT	0.00461	0.02091	**0.00008**	22.18	0.00101	0.00766	0.00006	7.09
	GK-Transformer	0.00398	0.01917	0.00013	25.21	0.00104	0.00744	0.00006	11.31
	Transolver	0.00403	0.01799	0.00011	33.91	0.00284	0.01352	0.00009	10.54
	HPNO(ours)	0.00258	**0.01135**	**0.00008**	57.71	**0.00066**	**0.00446**	**0.00004**	**5.38**
Foil	U-Net	0.00382	0.01337	0.00016	217.75	**0.00091**	0.00306	**0.00007**	16.93
	CNO	0.00288	0.00826	**0.00015**	126.51	0.00196	0.00582	0.00010	147.86
	DeepONet	0.00519	0.01273	0.00018	127.83	0.00357	0.01120	0.00019	123.67
	FNO	0.00370	0.01111	0.00017	143.63	0.00131	0.00450	0.00009	57.28
	WDNO	0.00260	0.00999	0.00023	**43.29**	0.00100	0.00344	0.00008	26.05
	MWT	0.00354	0.00878	**0.00015**	133.86	0.00131	0.00418	0.00009	28.65
	GK-Transformer	0.00376	0.01003	0.00016	158.20	0.00155	0.00455	0.00010	58.12
	Transolver	0.00429	0.01029	**0.00015**	72.25	0.00230	0.00593	0.00015	23.91
	HPNO(ours)	**0.00255**	**0.00647**	0.00016	61.02	0.00094	**0.00290**	**0.00007**	**15.31**
Combustion	U-Net	0.00536	0.01027	–	95.44	0.00254	**0.00547**	–	24.98
	CNO	0.00596	0.01406	–	123.45	0.00303	0.00657	–	35.51
	DeepONet	0.00561	0.01218	–	107.32	0.00284	0.00618	–	31.59
	FNO	0.00517	0.01036	–	90.14	0.00269	0.00561	–	29.17
	WDNO	0.00610	0.01707	–	134.93	0.00440	0.00888	–	57.62
	MWT	0.00551	0.01039	–	95.93	0.00257	0.00548	–	27.25
	GK-Transformer	0.00575	0.01294	–	112.75	0.00310	0.00728	–	37.13
	Transolver	0.00611	0.01523	–	122.06	0.00504	0.01215	–	77.46
	HPNO(ours)	**0.00505**	**0.00968**	–	**84.04**	**0.00248**	0.00561	–	**23.99**

E ADDITIONAL VISUALIZATIONS

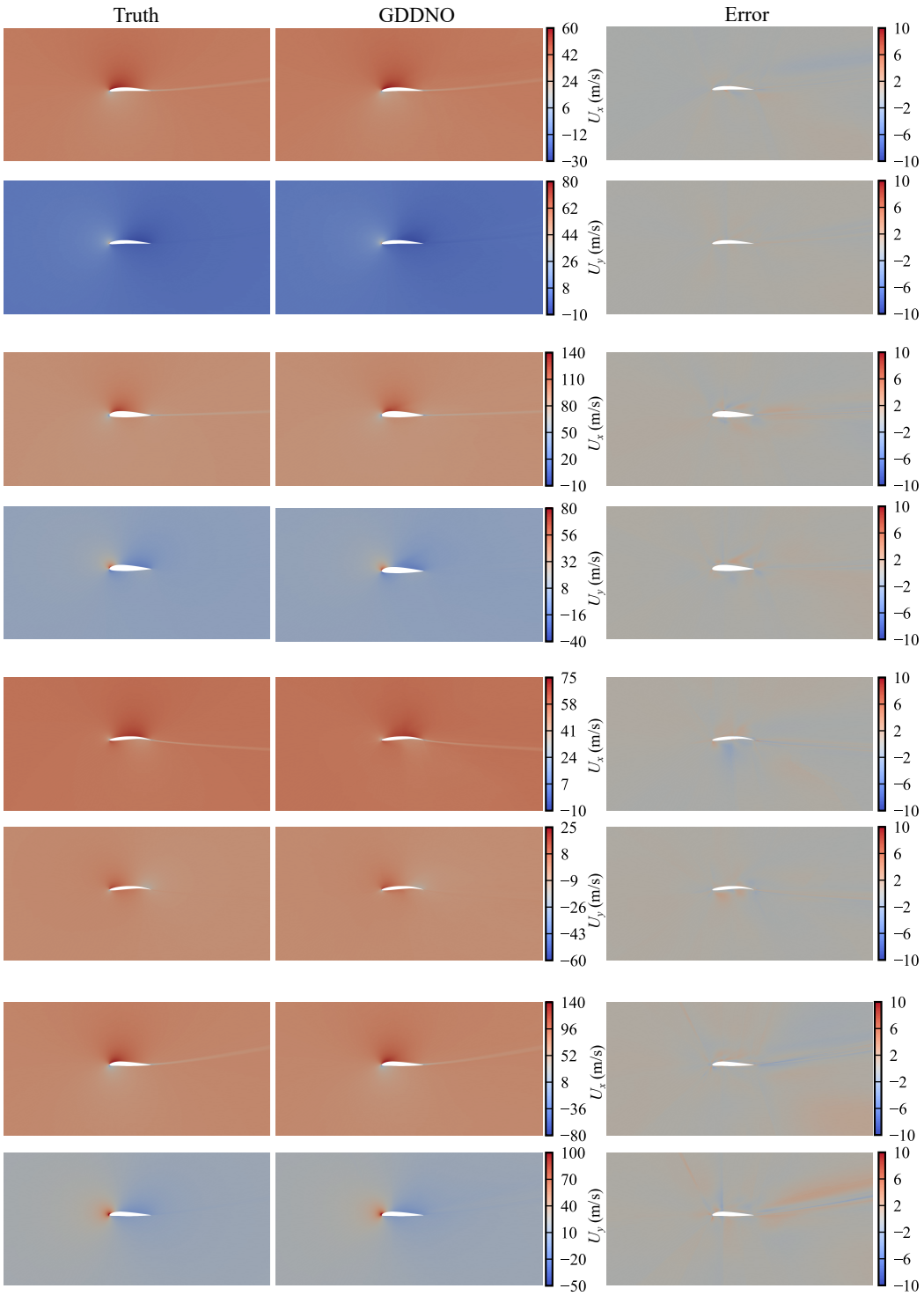

Figure 6: Calculation results and errors of different AirfRANS cases.

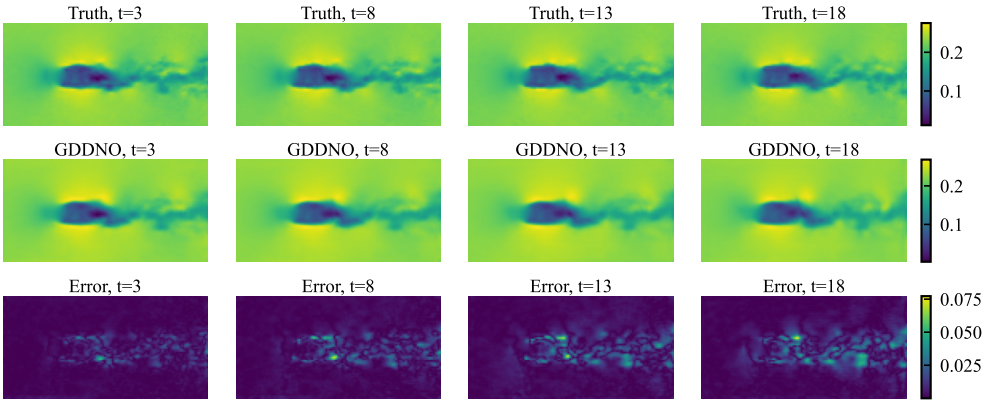

Figure 7: Visualization of results (u) on RealPDEBench cylinder.

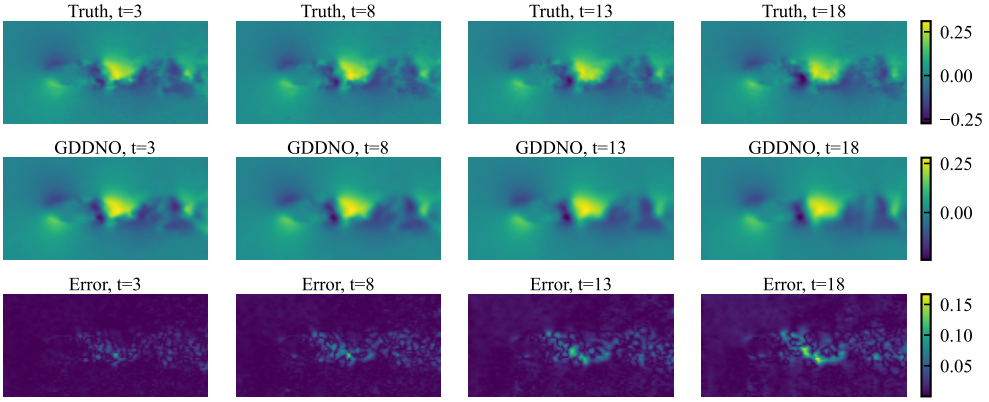

Figure 8: Visualization of results (v) on RealPDEBench cylinder.

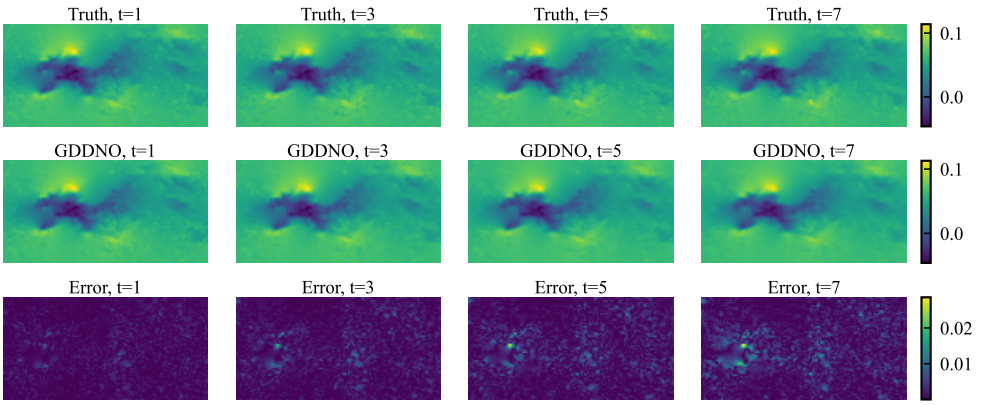

Figure 9: Visualization of results (u) on RealPDEBench controlled cylinder.

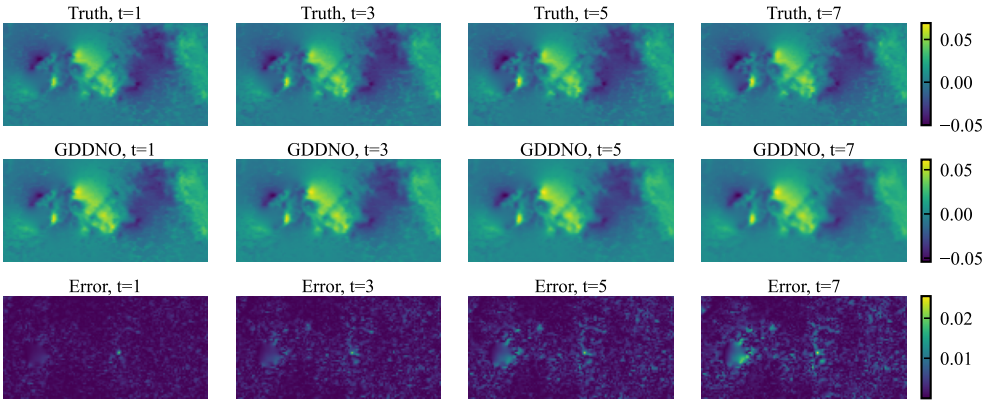

Figure 10: Visualization of results (v) on RealPDEBench controlled cylinder.

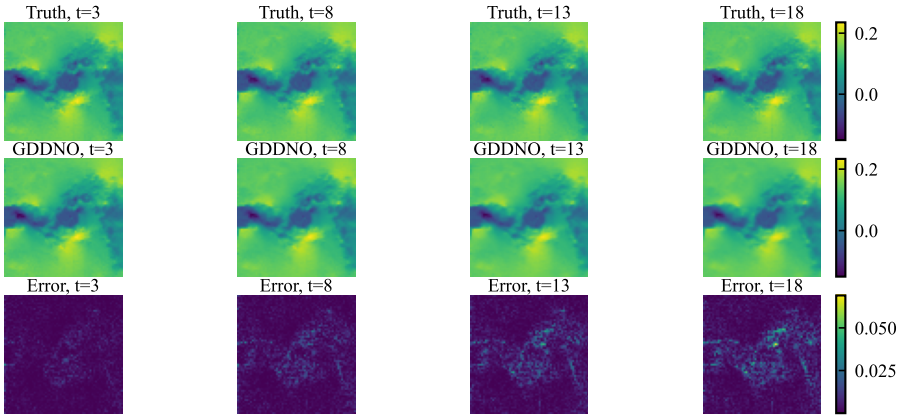

Figure 11: Visualization of results (u) on RealPDEBench FSI.

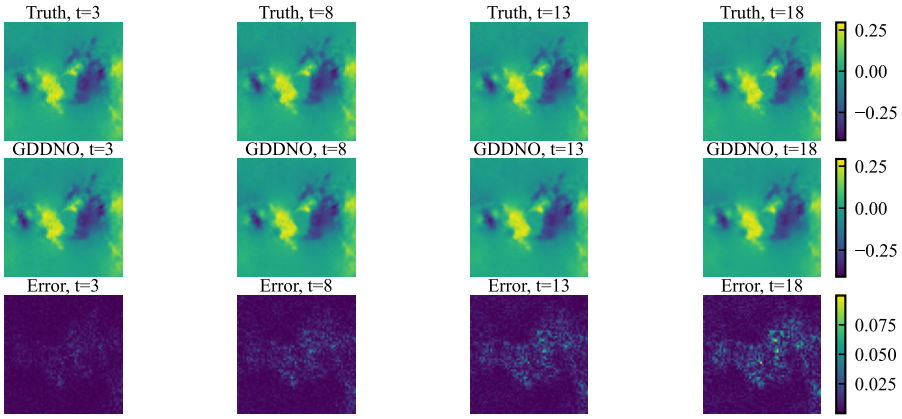

Figure 12: Visualization of results (v) on RealPDEBench FSI.

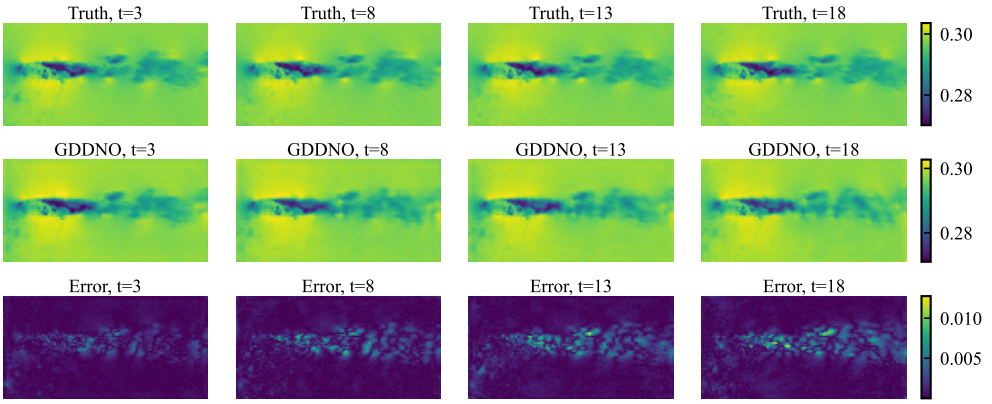

Figure 13: Visualization of results (u) on RealPDEBench foil.

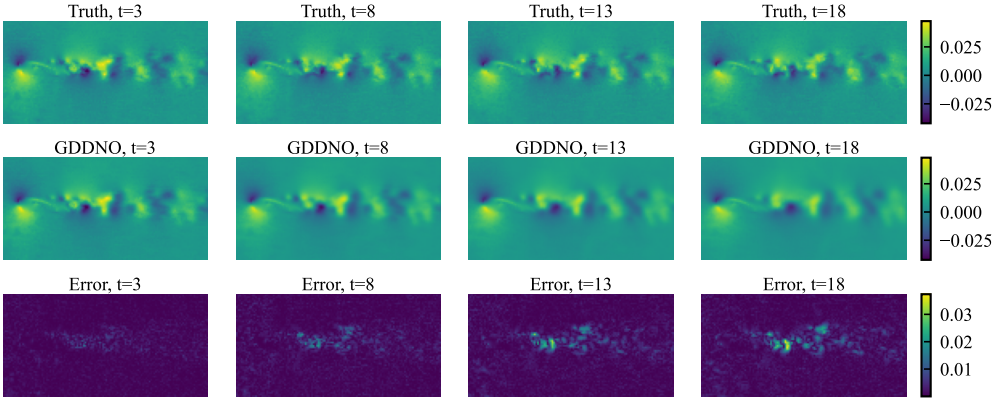

Figure 14: Visualization of results (v) on RealPDEBench foil.

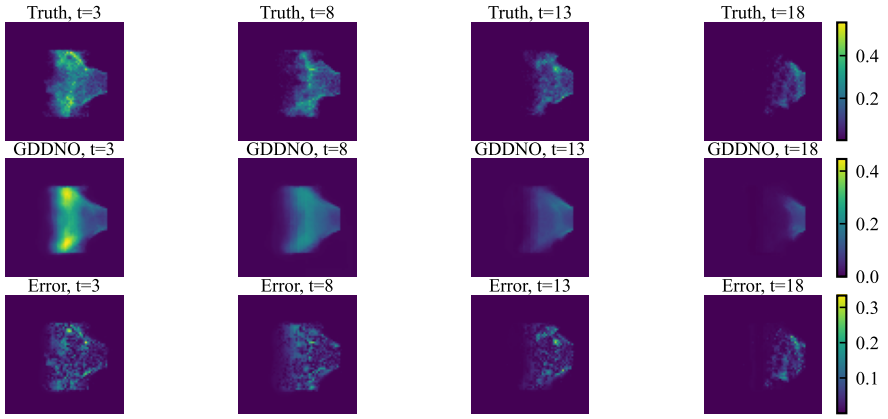

Figure 15: Visualization of results (I) on RealPDEBench combustion.

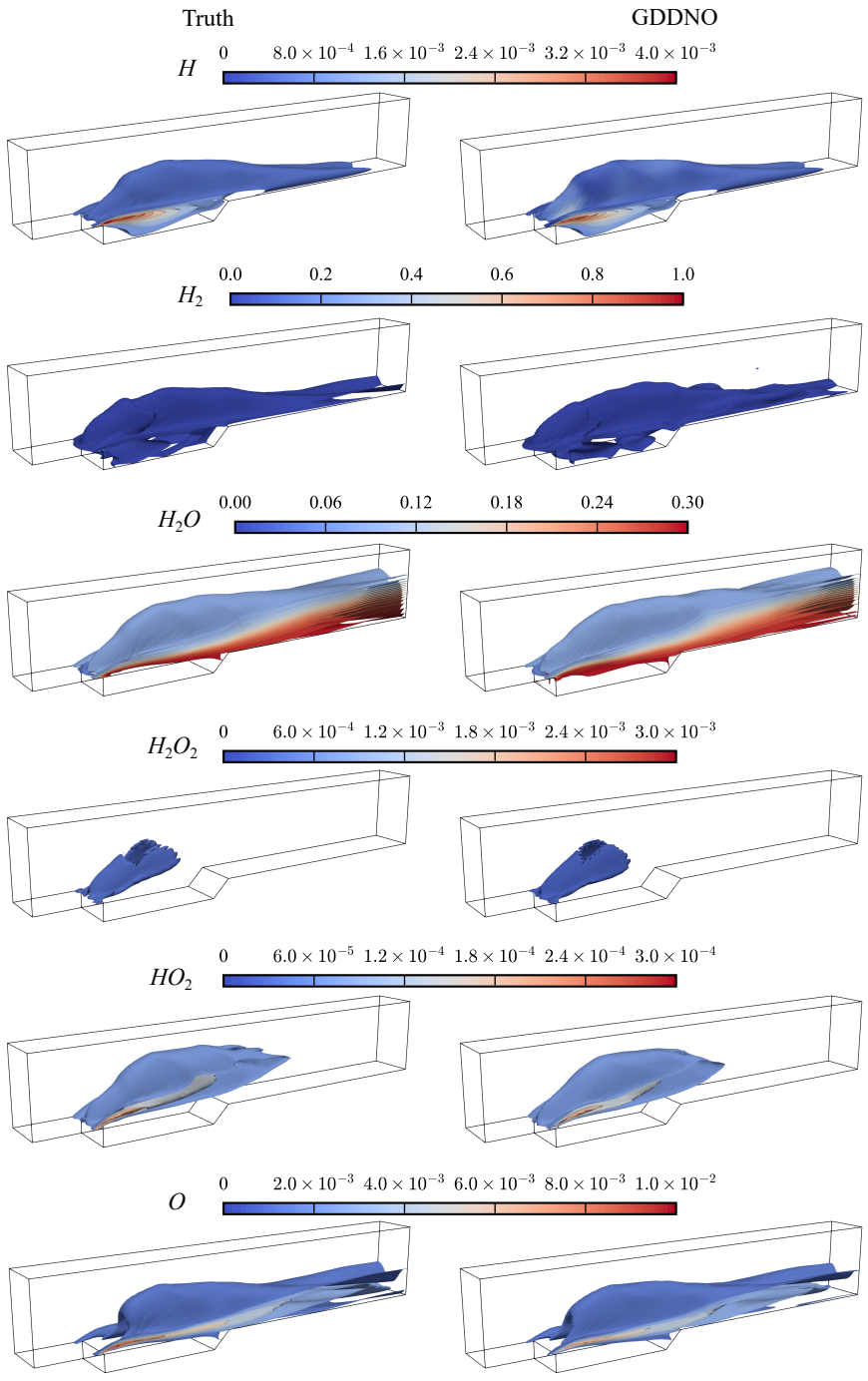

Figure 16: Visualization of H, H_2, H_2O, H_2O_2, HO_2, and O on cavity combustion, 3D results are presented as isosurfaces.

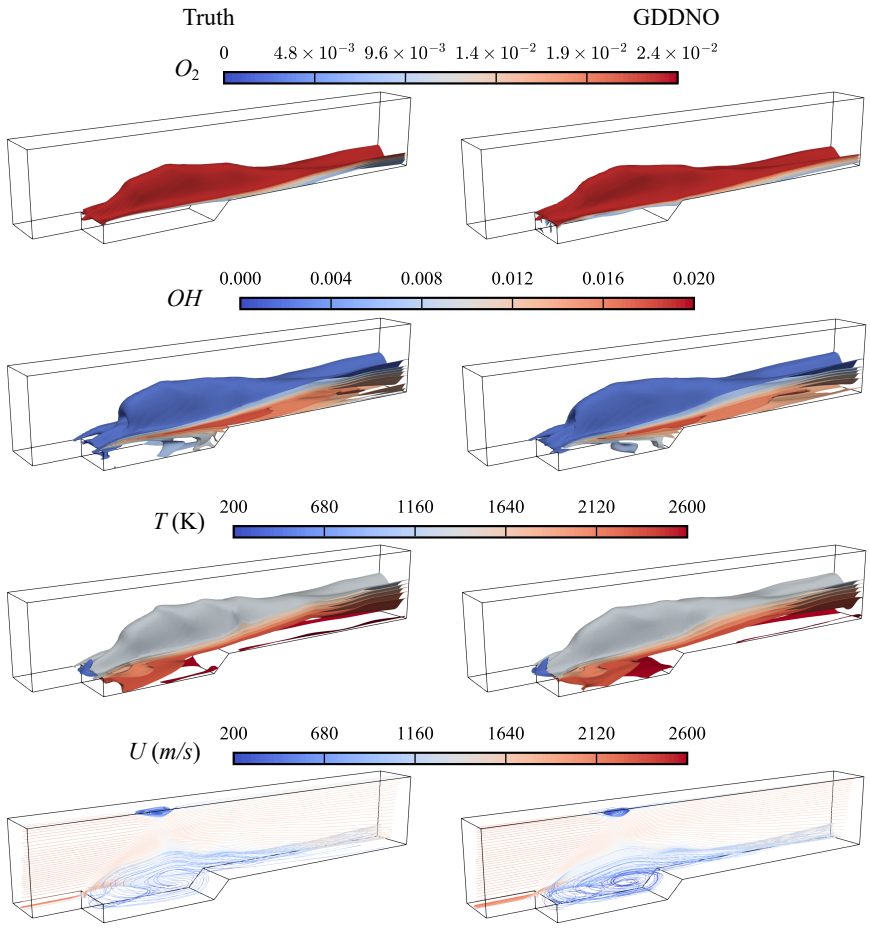

Figure 17: Visualization of O_2, OH, T, and U on cavity combustion, 3D results are presented as isosurfaces and velocity streamline.