Donglin CHEN, Xiang GAO, Chuanfu XU, Siqi WANG, Shizhao CHEN, Jianbin FANG, Zheng WANG, 2022. FlowDNN: a physics-informed deep neural network for fast and accurate flow prediction. *Frontiers of Information Technology & Electronic Engineering,* 23(2):207-219. https://doi.org/10.1631/FITEE.2000435

FlowDNN: a physics-informed deep neural network for fast and accurate flow prediction

Key words: Deep neural network; Flow prediction; Attention mechanism; Physics-informed loss

Corresponding author: Chuanfu XU
E-mail: xuchuanfu@nudt.edu.cn
ORCID: https://orcid.org/0000-0002-4876-2368

Motivation

1. For flow-related design optimization problems, e.g., aircraft and automobile aerodynamic design, computational fluid dynamics (CFD) simulations are commonly used to predict flow fields and analyze performance.

2. While important, CFD simulations are a resource-demanding and time-consuming iterative process.

3. The high simulation overhead limits the opportunities for large design space exploration and prevents interactive design.

Main idea

1. A novel deep neural network (DNN) to efficiently learn flow representations from CFD results named FlowDNN is proposed. It reduces the computation time by directly predicting the expected flow fields based on the given flow conditions and geometry shapes.

2. FlowDNN is the first DNN that incorporates the underlying physical conservation laws of fluid dynamics with a carefully designed attention mechanism for steady flow prediction.

3. This approach not only improves the prediction accuracy, but also preserves the physical consistency of the predicted flow fields, which is essential for CFD.

Method

1. To predict flow fields over different objects with deep networks, we first need to have an appropriate way to represent the object's geometric and domain boundaries. The lattice Boltzmann method (LBM) simulation inspires us to use artificial images to represent flow fields and boundaries, and to transform the flow field prediction into an image-to-image regression problem.

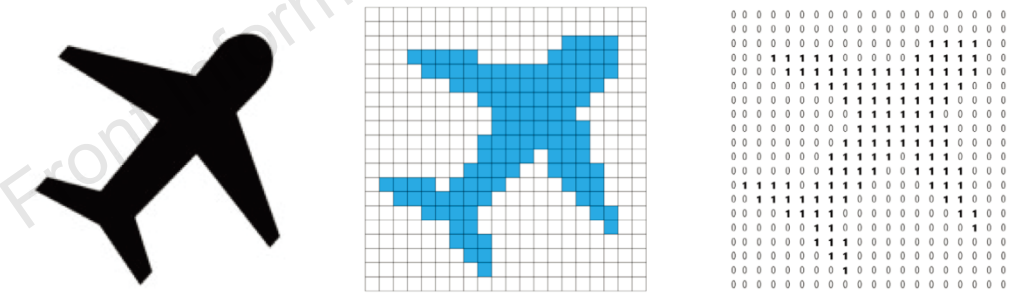


Fig. 1 Converting a 2D domain boundary to a Cartesian grid to generate a matrix input for our model

References to color refer to the online version of this figure

Method (Cont'd)

2. Network architecture. Our model takes input as a matrix that describes a two-dimensional (2D) geometry domain (of size 128×256 in this work) of the target object.

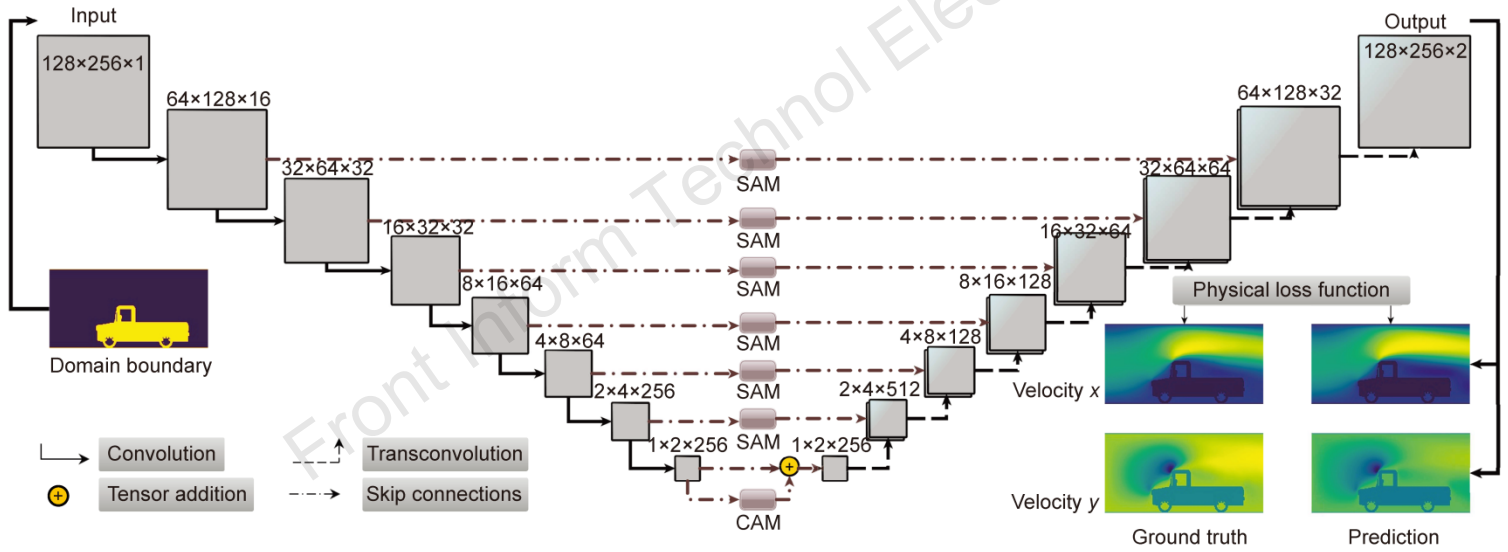


Fig. 2 The architecture of FlowDNN

The black arrows denote convolutional layers and transposed convolutional layers, while the brown arrows indicate the skip connections with the attention module (AM). The artificial image of the domain boundary is passed to the network as input. The output is the prediction of a 2D velocity field and is compared to the ground-truth data with the physical loss function. References to color refer to the online version of this figure

Method (Cont'd)

3. Physical loss functions. Our approach explicitly provides prior physical conservation law information to the network to enable it to extract features that satisfy physical consistency.

$$L_1 = \frac{1}{2mn_xn_y}\sum_{l=1}^{m}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\left(\left|u_{ij}^{l}-\overline{u}_{ij}^{l}\right|+\left|v_{ij}^{l}-\overline{v}_{ij}^{l}\right|\right), \tag{6}$$

$$L_{\text{physical}} = \alpha_1 L_1 + \alpha_2 L_{\text{mass}} + \alpha_3 L_{\text{momentum}}, \tag{5}$$

$$L_{\text{mass}} = \frac{1}{m\left(n_x-2\right)\left(n_y-2\right)} \cdot \sum_{l=1}^{m}\sum_{i=2}^{n_x-1}\sum_{j=2}^{n_y-1}\left|\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)_{ij}^{l}-\left(\frac{\partial \overline{u}}{\partial x}+\frac{\partial \overline{v}}{\partial y}\right)_{ij}^{l}\right|. \tag{7}$$

$$\begin{aligned} L_{\text{momentum}} = & \frac{1}{m\left(n_x-2\right)\left(n_y-2\right)}\sum_{l=1}^{m}\sum_{i=2}^{n_x-1}\sum_{j=2}^{n_y-1}\left\{\left|\left[\left(\frac{\partial(uu)}{\partial x}+\frac{\partial\left(uv\right)}{\partial y}\right)_{ij}^{l}-\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)_{ij}^{l}\right]\right.\right. \\ & \left.-\left[\left(\frac{\partial(\overline{uu})}{\partial x}+\frac{\partial\left(\overline{uv}\right)}{\partial y}\right)_{ij}^{l}-\frac{1}{Re}\left(\frac{\partial^2 \overline{u}}{\partial x^2}+\frac{\partial^2 \overline{u}}{\partial y^2}\right)_{ij}^{l}\right]\right|+\left|\left[\left(\frac{\partial(vu)}{\partial x}+\frac{\partial\left(vv\right)}{\partial y}\right)_{ij}^{l}-\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)_{ij}^{l}\right]\right. \\ & \left.\left.-\left[\left(\frac{\partial(\overline{vu})}{\partial x}+\frac{\partial\left(\overline{vv}\right)}{\partial y}\right)_{ij}^{l}-\frac{1}{Re}\left(\frac{\partial^2 \overline{v}}{\partial x^2}+\frac{\partial^2 \overline{v}}{\partial y^2}\right)_{ij}^{l}\right]\right|\right\}. \end{aligned} \tag{8}$$

Method (Cont'd)

4. Channel and spatial attention modules. We introduce attention mechanisms to our learning framework. This is motivated by the observation that some regions of interest (RoIs) in fluid flows often contain more important and complicated information than others as flow quantities change rapidly.

$$\boldsymbol{F}_{\mathrm{c}} = \boldsymbol{M}_{\mathrm{c}}(\mathbb{F}) \otimes \mathbb{F}, \tag{10}$$

$$\boldsymbol{F}_{\mathrm{s}} = \boldsymbol{M}_{\mathrm{s}}(\mathbb{F}) \otimes \mathbb{F}, \tag{11}$$

$$\boldsymbol{M}_{\mathrm{c}}(\mathbb{F}) = \sigma(\mathrm{MLP}(\mathrm{GAP}(\mathbb{F})) + \mathrm{MLP}(\mathrm{GMP}(\mathbb{F}))), \tag{12}$$

$$\boldsymbol{M}_{\mathrm{s}}(\mathbb{F}) = \sigma\left(\mathrm{Conv}\left(\mathrm{GAP}_{\mathrm{c}}\left(\mathbb{F}\right) \oplus \mathrm{GMP}_{\mathrm{c}}\left(\mathbb{F}\right)\right)\right), \tag{13}$$

Major results

Table 2 compares the baselines to FlowDNN in terms of accuracy, inference runtime, and the parameter size. Without any further optimization, FlowDNN with our physical loss function greatly outperforms its counterparts.

Table 2 Comparing FlowDNN with different baseline models

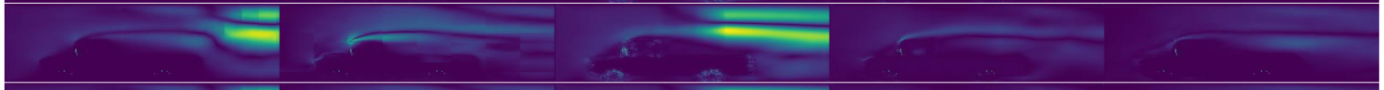
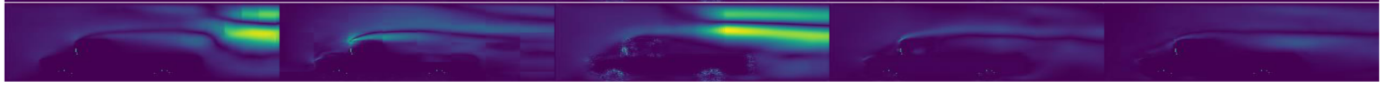
Method	MRE	MRE_{RoI}	MRE_{ma}	MRE_{mo}	Runtime (ms)	Parameter ($\times 10^6$)
LBM	–	–	–	–	3300×16	–
C-Net	14.31%	30.99%	37.44%	45.60%	9.29	180.25
T-Net	24.65%	59.71%	79.09%	82.38%	8.47	7.45
U-Net	14.74%	13.14%	30.78%	44.42%	16.15	32.96
FlowDNN	7.91%	23.56%	18.28%	22.46%	**3.52**	13.70
FlowDNN w/ AM	5.34%	9.16%	12.34%	15.69%	4.51	13.74
FlowDNN w/ AM and P	**4.77%**	**8.87%**	**12.14%**	**14.63%**	**3.62**	**7.40**

All FlowDNN models are trained using the physical loss function with AM and network pruning. AM: attention module; LBM: lattice Boltzmann method; MRE: mean relative error; RoI: region of interest; P: network pruning

Major results (Cont'd)

Table 3 quantifies the differences of some visualization samples of model predictions against the full-order CFD simulation results, indicating that our predictions are more visually closer to the results given by the full-order CFD solver.

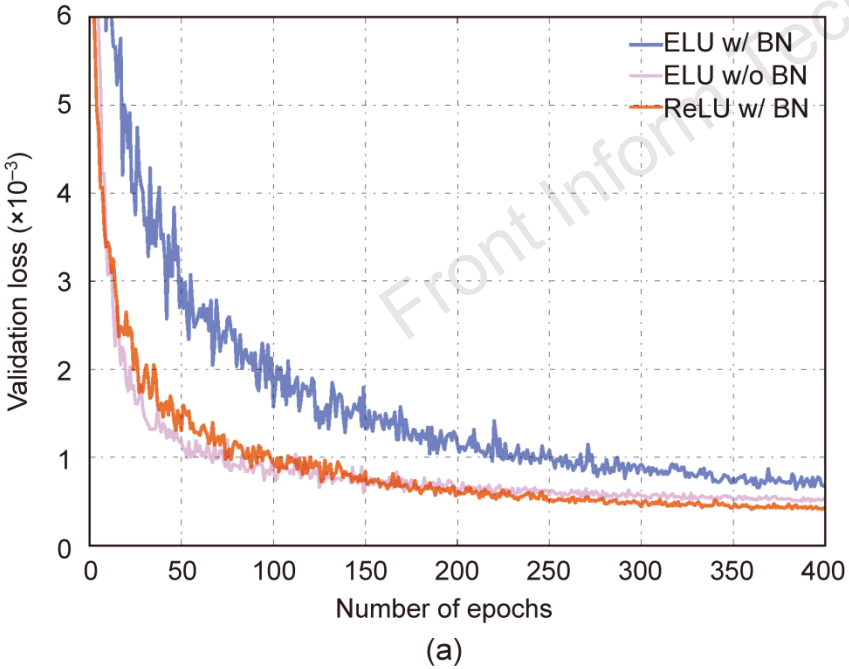
Table 3 The differences of baselines and FlowDNN compared with the ground truth

Vehicle type	U-Net	C-Net	T-Net	FlowDNN	FlowDNN AP
Racing					
Saloon					
Jeep					
Pickup					
Bus					

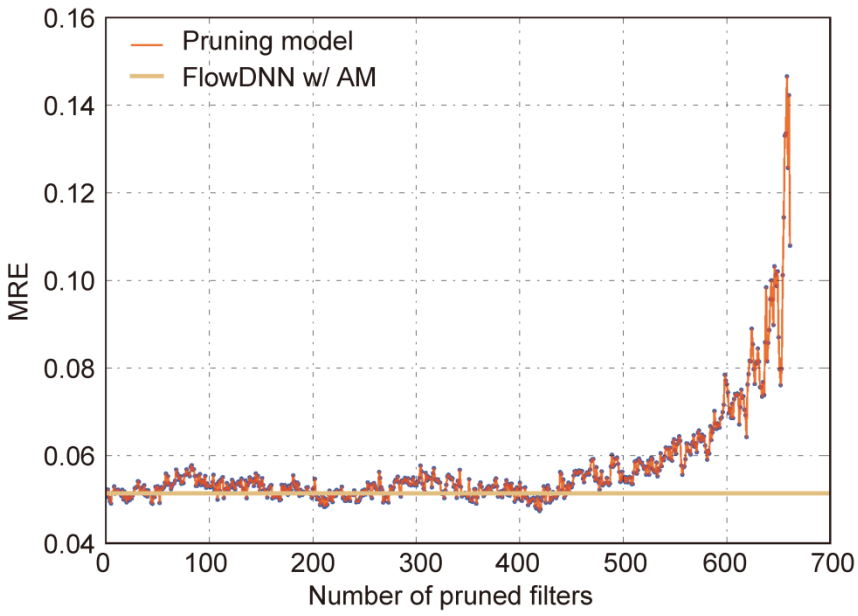
FlowDNN AP: FlowDNN with attention module and network pruning

Major results (Cont'd)

Fig. 8a shows that ReLU combined with batch normalization converges faster and delivers smaller errors than ELU on the validation dataset. Fig. 8b shows how the model scores along with network pruning.



(a)



(b)

Conclusions

1. We have presented FlowDNN, a novel DNN-based framework for predicting steady flow fields, to speed up full-order CFD simulations while preserving the physical conservation laws.

2. FlowDNN employs attention mechanism to learn better from the boundary layers.

3. It speeds up a GPU-accelerated CFD solver by over 14 000×.