Knowledge-Enhanced Deep Learning for Simulation of Tropical Cyclone Boundary-Layer Winds

Reda Snaiki, Teng Wu*

Department of Civil, Structural and Environmental Engineering, University at Buffalo, Buffalo, NY 14260, USA

**Corresponding author. Email:* tengwu@buffalo.edu

Abstract: Accurate and efficient modeling of the wind field is critical to effective mitigation of losses due to the tropical cyclone-related hazards. To this end, a knowledge-enhanced deep learning algorithm was developed in this study to simulate the wind field inside tropical cyclone boundary layer. More specifically, the machine-readable knowledge in terms of both physics-based equations and/or semi-empirical formulas was leveraged to enhance the regularization mechanism during the training of deep networks for dynamics of tropical cyclone boundary-layer winds. To comprehensively appreciate the high effectiveness of knowledge-enhanced deep learning to capture the complex dynamics using small datasets, two nonlinear flow systems governed respectively by 1D and 2D Navier-Stokes equations were first revisited. Then, a knowledge-enhanced deep network was developed to simulate tropical cyclone boundary-layer winds using the storm parameters (e.g., spatial coordinates, storm size and intensity) as inputs. The reduced 3D Navier-Stokes equations based on several state-of-the-art semi-empirical formulas were employed in the construction of deep networks. Due to the effective utilization of the prior knowledge on the tropical cyclone boundary-layer winds, only a relatively small number of training datasets (either from field measurements or high-fidelity numerical simulations) are needed. With the trained knowledge-enhanced deep network, it has been demonstrated that the boundary-layer winds associated with various tropical cyclones can be accurately and efficiently predicted.

***Keywords*:** Knowledge-enhanced deep learning, Tropical cyclones, Boundary-layer winds.

1. Introduction

Tropical cyclones, considered as warm-core and low-pressure systems, are responsible for inflicting severe economic losses and casualties through high wind, torrential rain and storm surge (e.g., Pielke et al. 2008; Czajkowski et al., 2011; Rappaport, 2014). This situation may become more complicated in light of changing climate and continued escalation of coastal population density (e.g., Snaiki and Wu 2018a). A mature tropical cyclone typically consists of a boundary-layer region, a region above the boundary layer with no radial motion, an updraft region, and a quiescent eye (Carrier et al. 1971). The high wind in the boundary layer is deemed as the most important component contributing to the tropical cyclone-related hazards.

The slab or depth-averaged wind model (e.g., Shapiro 1983; Vickery and Twisdale 1995; Smith and Vogl 2008) has been traditionally used in hurricane damage and loss estimation (e.g., Florida Hurricane Loss Projection Model and HAZUS-MH Hurricane Model) (Powell et al., 2005), where the Navier-Stokes equations are vertically averaged through a predefined boundary-layer depth. Since it assumes a constant boundary-layer height and estimates the surface wind speed based on empirical reduction factors, the depth-averaged wind model suffers from several shortcomings that may deteriorate the simulation accuracy (e.g., Kepert 2010a; 2010b). Hence, the height-resolving wind model has been treated as a superior alternative to the slab model (e.g., Khare et al. 2009; Kepert 2010a; Kepert 2010b). A simple and easy way to obtain the height-resolving winds is to use the empirical (or semi-empirical) vertical profiles recently developed with a large amount of measurement data both over ocean (e.g., using GPS dropsondes) and overland (e.g., using Weather Surveillance Doppler Radar) (Vickery et al. 2009; Snaiki and Wu 2018b). However, the use of predefined basis functions (e.g., polynomials, sines or cosines) in these vertical wind profiles may not be necessarily suitable for the accurate modeling of such a complex dynamic system. Over the past few decades, significant efforts have been made to acquire the height-resolving winds by directly solving the linear or nonlinear governing equations with a parametrization of the turbulent fluxes and surface drag (e.g., Kepert and Wang 2001). The simplicity and computational efficiency of the linear height-resolving wind models have made them widely implemented in various engineering applications (e.g., risk analysis) (Rosenthal 1962; Yoshizumi 1968; Meng et al. 1995; Kepert 2001; Huang and Xu 2012; Snaiki and Wu 2017a; 2017b; Fang et al. 2018). However, there remain important phenomena that the physics-based linear schemes are not able to properly address due to their inherent limitations (e.g., underestimation of supergradient winds). While the physics-based nonlinear height-resolving wind models based on the fully-order Navier Stokes equations (e.g., Weather Research and Forecasting model) present improved simulation accuracy, they are impractical for the real-time forecasting and not suited for risk assessment due to the high computational demands.

Alternatively, data-driven modeling or machine learning may be considered as an effective tool for rapid prediction of tropical cyclone boundary-layer winds. Machine learning techniques can be implemented to simulate nonlinear dynamic systems without committing to simplified assumptions or linearization. In particular, the artificial neural network (ANN), due partially to the solid mathematical foundation laid by Hornik et al. (1989) for a theorem stating that an ANN with a single layer of enough hidden units can approximate any multivariate continuous function with arbitrary accuracy, has been widely utilized in diverse scientific disciplines and applications during the last several decades. ANNs have also been applied to simulate the wind field inside tropical cyclones. Huang and Xu (2013) utilized the ANN to consider topography effects in the simulation of directional typhoon wind field at one location (i.e., Stonecutters Bridge in Hong Kong). Wei et al. (2018a) proposed an ANN model to predict the typhoon-induced wind speed at three stations in western Taiwan (i.e., Hsinchu Station, Wuqi Station and Kaohsiung Station) based on historical data. It is noted that both studies developed their ANN models for very limited localized points, and hence are unable to simulate the spatial distribution of boundary-layer winds inside the tropical cyclones. To enhance the simulation performance by more effectively exploring and finding hidden information from data of nonlinear systems, the standard neural networks have recently been advanced to deep neural networks (DNNs) along with the rapid developments of central processing unit (CPU) and graphics processing unit (GPU) computing tools (LeCun et al. 2015). A large amount of high-fidelity input-output datasets are typically needed during the training process of deep learning. Unfortunately, big datasets may not be always available for an engineering application due partially to the high cost of data generation (using numerical/experimental/field-measurement approach) (e.g., Swischuk et al. 2018).

To alleviate the high demand of high-fidelity training datasets for the classical neural networks, a hybrid system consisting of neural-network and first-principle components has been developed (e.g., Psichogios and Ungar 1992; Beidokhti and Malek 2009). In addition to significantly reducing the amount of data that the purely data-driven models require, the hybrid scheme presents better interpolation and extrapolation results than those from the standard neural networks due to its physical constraints (Psichogios and Ungar 1992). Deep reinforcement learning (DRL), combining both the DNNs and reinforcement learning properties, could be considered as an improved, modern version of the conventional hybrid system (e.g., Schmidhuber, 2015; Sutton and Barto 2018; Wei et al. 2018b). Through their critic functions, DRLs maximize the system rewards based on a trial-and-error approach to achieve the best outcomes. Using the governing equations along with the boundary and initial conditions as the critics to penalize the loss function and regularize the learning process, Raissi et al. (2017a; 2017b) recently proposed a data-efficient, physics-informed deep learning to solve nonlinear partial differential equations (PDEs). The automatic differentiation, a generalized backpropagation algorithm, was used to facilitate the minimization of the loss function in the physics-informed deep learning (Baydin et al. 2018; van

Merriënboer et al. 2018). It is noted that the physics-informed deep learning is unable to simulate the tropical cyclone boundary-layer winds governed by the Navier-Stokes equations at high Reynolds numbers.

In this study, a more general knowledge-enhanced deep learning algorithm will be developed to predict the spatial distribution of tropical cyclone boundary-layer winds with high computational efficiency and simulation accuracy. The reduced Navier-Stokes equations based on several state-of-the-art semi-empirical formulas are employed as part of the loss function in deep learning (fully connected, feedforward system) to provide machine-readable prior knowledge that facilitates the effective regularization of the neural networks. Unlike the data-driven solution/discovery applications developed by Raissi et al. (2017a; 2017b), essential parameters of the complex dynamic system are treated here as the network inputs. Accordingly, the developed knowledge-enhanced deep learning can be used to predict the spatial distribution of tropical cyclone boundary-layer winds for an arbitrary scenario with standard storm parameters available to public (e.g., provided by the National Hurricane Center), namely hurricane center location, central pressure difference, radius of maximum winds, translational speed, approach angle and surface roughness. The relatively small number of datasets needed for the training of the proposed knowledge-enhanced deep learning are generated using a recently developed hurricane wind model (Snaiki and Wu 2019, see Appendix A). It is demonstrated that the trained knowledge-enhanced deep network provides accurate and efficient simulations of the boundary-layer winds inside tropical cyclones, and hence can be utilized as an effective tool for wind hazard prediction as part of the early warning system.

2. Theoretical Background

2.1 Deep neural network

Machine learning technique is a form of artificial intelligence, first inspired by the human biological nervous system, designed to establish complex mathematical relationships between input and output data (e.g., McCulloch and Pitts 1943; Hornik et al. 1989; Ghosh and Shin 1992; Haykin 1994). The state-of-the-art machine learning algorithms (e.g., deep convolutional and recurrent networks) have exhibited outstanding performances in capturing hidden patterns from the data as well as any inherent nonlinearities of the system. The recent success of deep neural networks (DNNs) is due mainly to the significant increase of large databases and computational power (e.g., Kutz 2017). A typical DNN structure consists of an input layer, an output layer and a number of hidden layers, as illustrated by Fig. 1. Each hidden layer is composed of multiple artificial neurons, the most fundamental elements in the DNN architecture. The output y of an artificial neuron can be obtained through the following formula:

$$y = f\left(b + \sum_i w_i x_i\right) \tag{1}$$

where f = activation function (e.g., sigmoidal function, exponential linear unit, hyperbolic tangent function or rectified linear unit); b = bias; w_i = weight associated with input x_i. The weights and biases are calibrated during the training process based on the available input-output datasets.

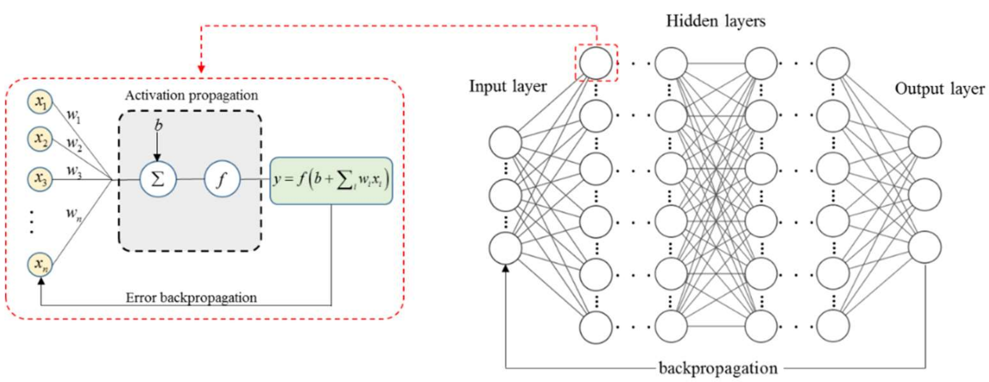

Fig. 1. Structure of artificial neurons used in DNNs

Two essential operations are typically necessary for the DNN training, namely activation propagation (feedforward) and error propagation (back-propagation) as shown in Fig. 1. During the first stage, each neuron delivers an output signal based on Eq. (1). The resulting output is then compared to the target one, and the obtained errors are backpropagated through the network during the second stage. The weights and biases are adjusted based on a selected optimizer algorithm (e.g., gradient-descent method) to minimize the predefined loss function (objective function) (Rumelhart et al. 1986; LeCun et al. 1989; 2012). Multiple iterations are needed during the supervised training process until a convergence criteria is satisfied. Advanced optimizer models (e.g., stochastic gradient-descent approach or Adam optimizer) have been developed in the literature to overcome the local minima convergence and slow learning rate of the standard backpropagation algorithm (Gurner 1997; Atakulreka and Sutivong 2007; Constantinescu et al. 2008; Ruder 2016).

The hyperparameters related to both model structure (e.g., number of inputs, number of layers, number of neurons per layer, activation function and weight initialization) and optimizer (e.g., learning rate and number of training iterations for the gradient-descent algorithm) should be carefully selected and tuned since they present strong influence on the training efficiency and network performance (e.g., Wu and Kareem 2011; Thornton et al. 2012; Bardenet et al. 2013). Due to a lack of general rules for the determination of optimum model structure, the tedious trial-and-error approaches are usually adopted for each particular problem. Once the training phase is completed, the DNN provides output of the simulated system through a simple arithmetic operation with any desired input information and hence circumvents the extreme computational cost of classical numerical methods (e.g., finite element or finite difference).

2.2 Knowledge-enhanced deep learning

One challenge in the applications of DNNs to many engineering problems is that the large number of high-quality data needed for the training purpose may not be available. In addition, the fact of being merely black box makes the DNN not easy to reasonably interpret and to accurately interpolate/extrapolate. Hence, research efforts have been made to open the black box of DNN by integrating the prior knowledge of the target system into the model development. Accordingly, a hybrid system consisting of neural-network and first-principle components has been developed (e.g., Psichogios and Ungar 1992; Beidokhti and Malek 2009). In the context of DRL, Raissi et al. (2017a; 2017b) proposed a data-efficient, physics-informed deep learning as an improved, modern version of the conventional hybrid scheme to solve nonlinear PDEs. In such a case, the loss function is penalized using the governing differential equations as constraints (Raissi et al. 2017a; 2017b). In addition to the rationalism-based knowledge, the empiricism-based knowledge is also critical in an engineering setting. Hence, both the physics-based equations and the semi-empirical (or purely empirical) formulas are leveraged in this study together with the available numerical/experimental/field-measurement training data to form a more general knowledge-enhanced deep learning. Figure 2 presents a schematic of the proposed knowledge-enhanced deep learning, where x is system input coordinate; u' is system solution; N_x^p is rationalism-based system operator; N_x^e is empiricism-based system operator; and λ is system parameters.

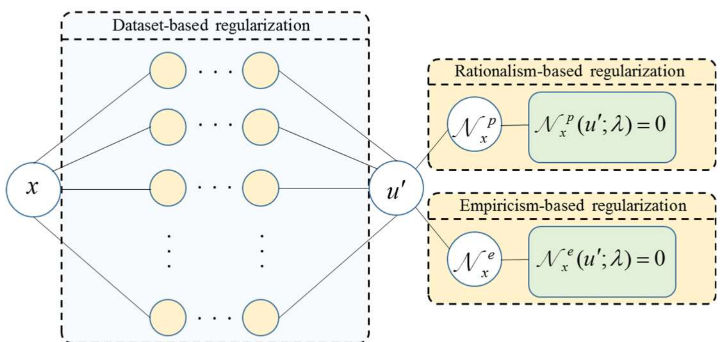

Fig. 2. Schematic of knowledge-enhanced deep learning

Accordingly, the total loss function of the knowledge-enhanced deep learning is expressed as:

$$L(\sigma) = \frac{1}{N_d}\sum_{j=1}^{N_d}\left[u'(x_d^j)\Big|_\sigma - u(x_d^j)\right]^2 + \frac{1}{N_p}\sum_{j=1}^{N_p}\left[F_x^{'}(x_p^j;\lambda)\right]^2\Big|_\sigma + \frac{1}{N_e}\sum_{j=1}^{N_e}\left[G_x^{'}(x_e^j;\lambda)\right]^2\Big|_\sigma \quad (2)$$

where $\left\{x_d^j, u(x_d^j)\right\}_{j=1:N_d}$ are available training datasets; $U' := \left\{x_d^j, u'(x_d^j)\right\}_{j=1:N_d}$ is data-driven-based network; $F_x^{'} := \left\{ x_p^j, N^{\ p}[u'(x_p^j), \lambda] = 0 \right\}_{j=1:N_p}$ is rationalism-based network; $G_x^{'} := \left\{ x_e^j, N^{\ e}[u'(x_e^j), \lambda] = 0 \right\}_{j=1:N_e}$ is empiricism-based network; N_d , N_p and N_e are numbers of available training datasets, sampled points from physics-based equations and sampled points from semi-empirical (or empirical) formulas ($N_f = N_p + N_e$), respectively; and σ = network weights and biases to be determined during the training process. Figure 3 illustrates the general algorithm of the knowledge-enhanced deep learning to approximate a given nonlinear system. According to Fig. 3, the automatic differentiation is employed to compute the necessary derivatives of the output with respect to the input in the physics-based equations and/or semi-empirical formulas through the chain rule for effectively embedding prior knowledge of the simulated system. This technique is readily available in several machine learning packages such as Tensorflow (Abadi et al. 2016) used in this study. As one of the most popular and widely-used, open-source libraries for machine learning and high performance numerical computation, Tensorflow was originally developed by Google in C++ with a python interface (Abadi et al. 2016).

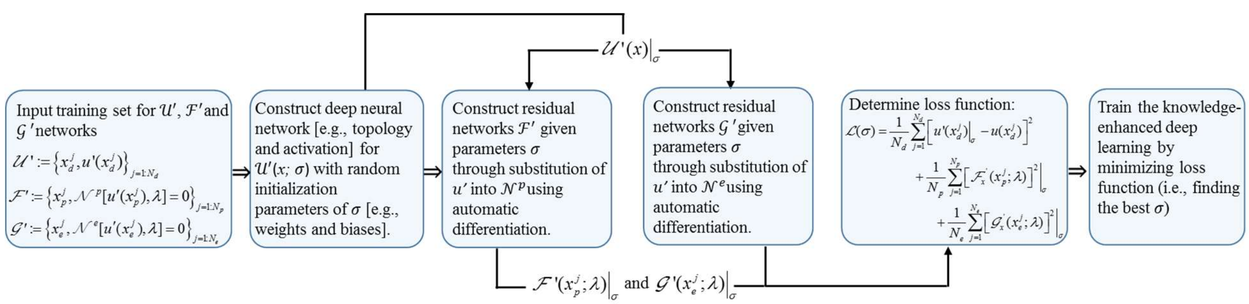

Fig. 3. Knowledge-enhanced deep learning algorithm

3. Knowledge-Enhanced Deep Learning for 1D and 2D Flow Simulations

To comprehensively appreciate the high effectiveness of knowledge-enhanced deep learning with various network structure (i.e., number of layers and number of neurons per each layer) to capture the complex dynamics using small datasets, two nonlinear flow systems, namely Burgers viscous flow and Taylor-Green vortex flow governed respectively by 1D and 2D Navier-Stokes equations are revisited herein. The first example flow has been recently studied in a deep-learning setting (e.g., Raissi et al. 2017a, Wei et al. 2018b), however, only a unique value of kinematic viscosity was investigated. The viscosity is an important parameter that significantly changes the solution of Burgers equation, hence, it will be treated here as a network input in the knowledge-enhanced deep learning. Special focus will be given to the low viscosity values corresponding to high Reynolds numbers. While the effects of the sampled points from all governing equations N_f (together with initial and boundary training data N_i and N_b , respectively) on the model performance will be evaluated based on the first case study, the effects of the available training data size N_d on the model predictive accuracy will be highlighted using the second case study of Taylor-Green vortex flow. The loss minimization will be achieved through the L-BFGS-B optimization algorithm due to its superior rate of convergence for a wide range of physical problems (Liu and Nocedal 1989; Byrd et al. 1995), together with the popular Xavier's normal initialization scheme (Glorot and Bengio 2010) to properly initialize the weights of the deep neural network. The model accuracy results will be investigated on the L_2-norm basis.

3.1 One-dimensional Burgers viscous flow

The Burgers viscous flow is governed by 1D Navier-Stokes equation without the pressure gradient term (Burgers 1948):

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}-\upsilon\frac{\partial^2 u}{\partial x^2}=0 \tag{3}$$

where υ = kinematic viscosity; $u(t,x,\upsilon)$ = fluid velocity at the time *t* and location *x* ($0\le t\le 1$ and $-1\le x\le 1$). In this case study, the initial condition of $u(0,x,\upsilon)=-\sin(\pi x)$ and boundary condition of $u(t,-1,\upsilon)=u(t,1,\upsilon)$ were selected to solve Eq. (3). The solution $u(t,x,\upsilon)$ is approximated by a knowledge-enhanced deep network with a hyperbolic tangent activation function. The total cost function (mean squared error loss) is then expressed as:

$$L = \frac{1}{N_i}\sum_{j=1}^{N_i}\left|u'(t_i^j, x_i^j, \upsilon_i^j) - u_i^j\right|^2 + \frac{1}{N_b}\sum_{j=1}^{N_b}\left|u'(t_b^j, x_b^j, \upsilon_b^j) - u_b^j\right|^2 + \frac{1}{N_f}\sum_{j=1}^{N_f}\left|f(t_f^j, x_f^j, \upsilon_f^j)\right|^2 \quad (4)$$

where $\{t_i^j, x_i^j, \upsilon_i^j, u_i^j\}$ = initial training data; $\{t_b^j, x_b^j, \upsilon_b^j, u_b^j\}$ = boundary training data; $\{t_f^j, x_f^j, \upsilon_f^j\}$ = collocation points with $f = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \upsilon\frac{\partial^2 u}{\partial x^2}$ and N_i, N_b and N_f are corresponding numbers of randomly distributed spatial-temporal datasets. The generalized parametric viscous Burgers equation was also numerically solved using the conventional spectral methods for comparison purpose (denoted as reference solution). More specifically, the solution was obtained using the Chebfun package (Driscoll et al. 2014), an open source software package that can be implemented with Matlab, with a spectral Fourier discretization of 512 modes and a fourth-order explicit Runge-Kutta time integration. Equation (4) was integrated up to the final time $t = 1$ with a time step of 10^{-4}. The comparison of the spatio-temporal evolution of $u(t, x, \upsilon)$ between the reference and simulated solutions is presented in Fig. 4 for three different values of kinematic viscosity, namely 0.003, 0.032 and 0.064. The employed baseline knowledge-enhanced deep learning possesses one input layer (with 3 inputs $\{t, x, \upsilon\}$), four hidden layers (each with 25 neurons) and one output layer (with 1 output) (i.e., network architecture of 3-25-25-25-25-1) with $N_i = 1000$, $N_b = 1000$ and $N_f = 20000$. The comparison was also carried out at two different time instants (i.e., $t = 0.25$ and $t = 0.75$), as shown in Fig. 5. It is noted that excellent agreement between the reference and simulated solutions was achieved and the relative L_2-norm value between the simulated and reference solutions is less than 10^{-3}. It is noted that the increase of Reynolds number (decrease of viscosity) facilitates the development of a shock profile around $x = 0$.

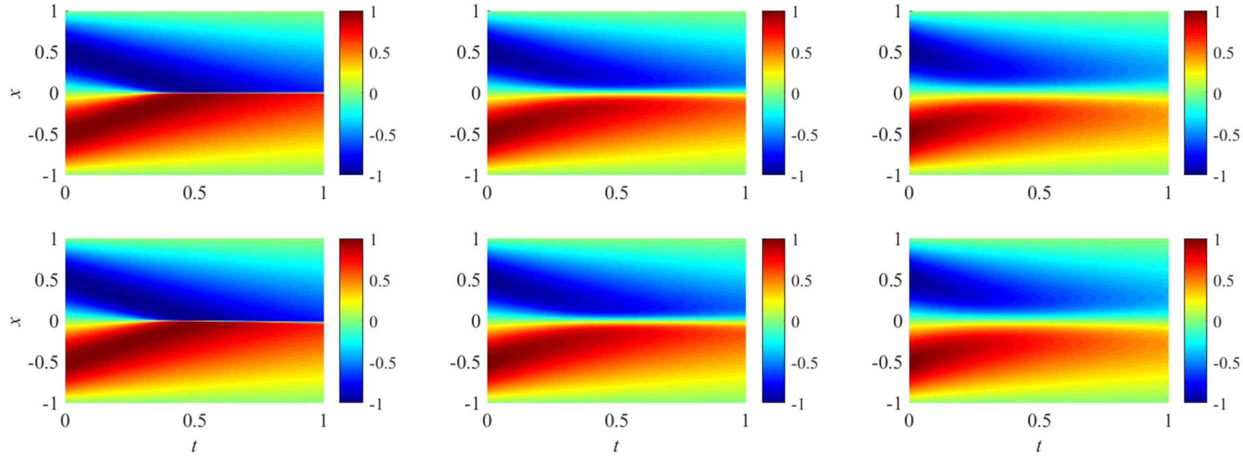

Fig. 4. Comparison between reference (top) and simulated (lower) solutions of Burgers viscous flow for three different kinematic viscosity values: $\upsilon = 0.003$ (left), $\upsilon = 0.032$ (middle) and $\upsilon = 0.064$ (right)

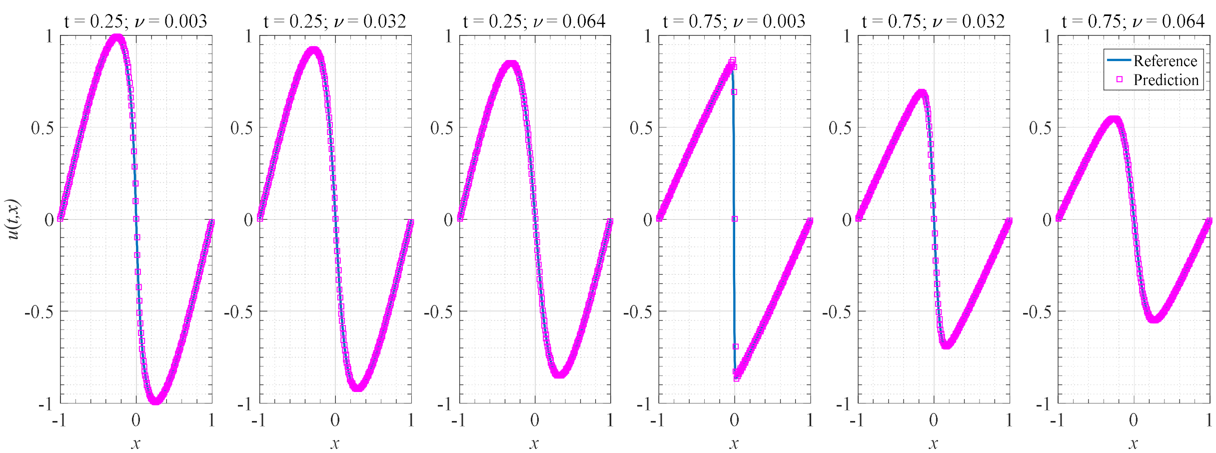

Fig. 5. Comparison between reference and simulated solutions of Burgers viscous flow at $t = 0.25$ and $t = 0.75$ for various kinematic viscosity values

To investigate the effects of network architecture and sampled points from the governing equation N_f (together with N_i and N_b) on the model performance, Fig. 6(a) reports the normalized L_2-norm values (by the highest L_2-norm value) for various number of hidden layers and neurons per each layer with $N_i = 1000$, $N_b = 1000$ and $N_f = 10000$. Clearly, the normalized L_2-norm values decrease (and hence the simulation accuracy increases) with the number of layers and neurons. Figure 6(b) presents the normalized L_2-norm for various N_i, N_b and N_f with a network architecture of 3-25-25-25-25-1. It is noted that the prediction accuracy generally increase with N_f (as well as N_i or N_b), indicating that the prior knowledge of the flow system in terms of governing equations enhances simulation fidelity of the deep neural network.

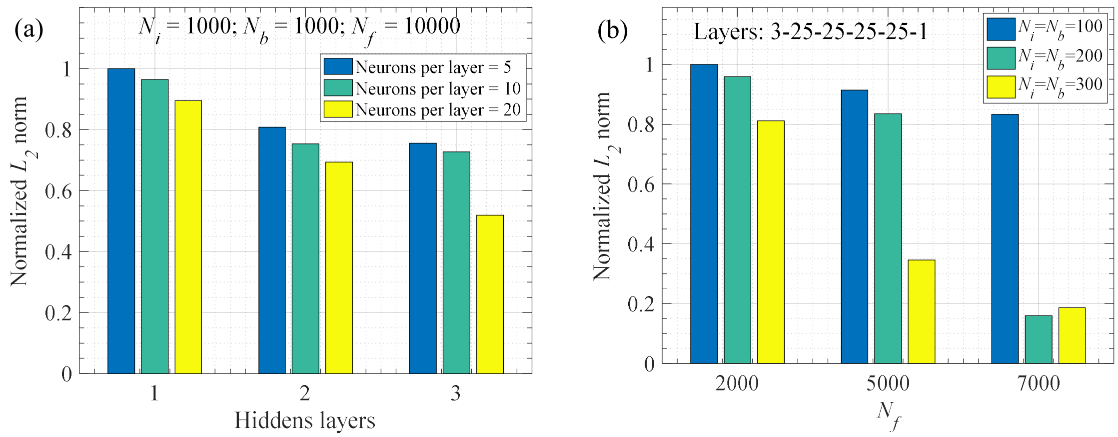

Fig. 6. Normalized L_2-norm values for simulation of Burgers viscous flow: (a) various network architectures and (b) various datasets

3.2 Two-dimensional Taylor-Green vortex flow

The Taylor-Green vortex flow is governed by the 2D incompressible Navier-Stokes equations together with a continuity equation (Brachet et al. 1983; Canuto et al. 2007):

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+\upsilon\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{5a}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+\upsilon\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right) \tag{5b}$$

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \tag{5c}$$

where (u,v) = fluid velocity components; p = pressure; and ρ = fluid density. With the imposed initial and periodic boundary conditions in both x and y directions, the exact closed-form solution (denoted as reference solution) of Taylor-Green vortex in a squared domain ($0 \le x, y \le 2\pi$) is given as:

$$u(x,y,t)=\cos(x)\sin(y)\exp(-2\upsilon t) \tag{6a}$$

$$v(x,y,t)=-\sin(x)\cos(y)\exp(-2\upsilon t) \tag{6b}$$

In addition, the pressure is given as $p = \frac{-\rho}{4}[\cos(2x) + \cos(2y)]\exp(-4t)$. The solutions $u(x, y, t)$ and $v(x, y, t)$ with $\upsilon = 1$ are approximated by a knowledge-enhanced deep learning with a hyperbolic tangent activation function and a network architecture of 3-50-50-50-50-2. The total cost function is then expressed as:

$$L = \frac{1}{N_i}\sum_{j=1}^{N_i}\left\{\left|u'(t_i^j, x_i^j) - u_i^j\right|^2 + \left|v'(t_i^j, x_i^j) - v_i^j\right|^2\right\} + \frac{1}{N_b}\sum_{j=1}^{N_b}\left\{\left|u'(t_b^j, x_b^j) - u_b^j\right|^2 + \left|v'(t_b^j, x_b^j) - v_b^j\right|^2\right\} + \frac{1}{N_f}\sum_{j=1}^{N_f}\left\{\left|f_1(t_f^j, x_f^j)\right|^2 + \left|f_2(t_f^j, x_f^j)\right|^2 + \left|f_3(t_f^j, x_f^j)\right|^2\right\} \quad (7)$$

where f_1, f_2 and f_3 are defined as $f_1 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$, $f_2 = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} - \upsilon\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$ and $f_3 = \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} - \upsilon\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right)$, respectively. The numbers of randomly distributed spatio-temporal points in this example are $N_i = 600$, $N_b = 600$ and $N_f = 50000$, respectively. The comparison of the spatial evolution of solutions $u(x, y, t)$ and $v(x, y, t)$ at $t = 0.1$ in terms of the total component $U = \sqrt{u^2 + v^2}$ between the reference and simulated solutions is presented in Fig. 7. It is noted that excellent agreement between the reference and simulated solutions was achieved, and the relative L_2-norm value (between the simulated and reference solutions) is less than 10^{-3}.

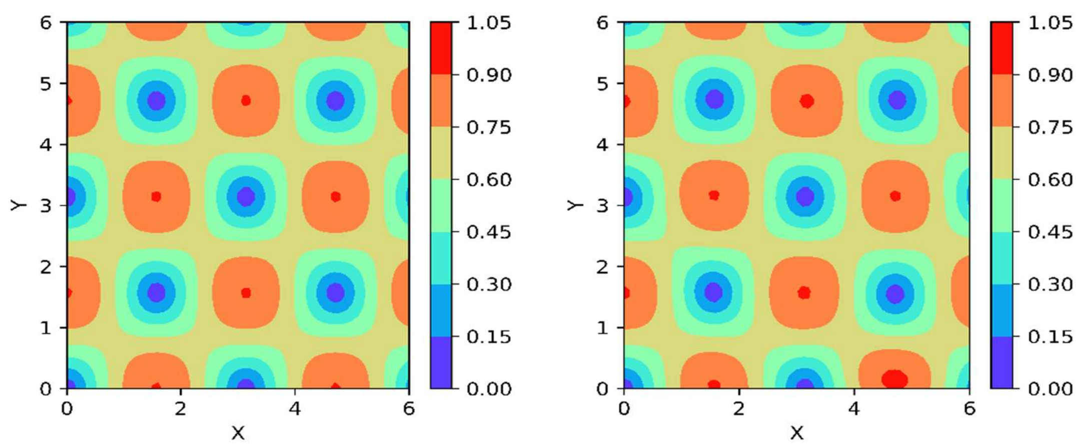

Fig. 7. Comparison of reference (left) and simulated (right) solutions of Taylor-Green vortex flow

To investigate the effects of available training data retrieved from the closed-form solution of Eqs. (6a) and (6b) in this study on the performance of knowledge-enhanced deep network, Fig. 8 reports the normalized L_2-norm values (by the highest L_2-norm value) for various sizes of N_d. It is noted that the normalized L_2-norm values decrease (and hence the simulation accuracy increases) with the number of training points. More specifically, the simulation accuracy has increased up to around 15% by introducing the computational training points. This observation actually indicates that the knowledge-enhanced deep learning can be effectively trained using small datasets from numerical/experimental/field-measurement efforts, along with a large number of datasets from prior knowledge.

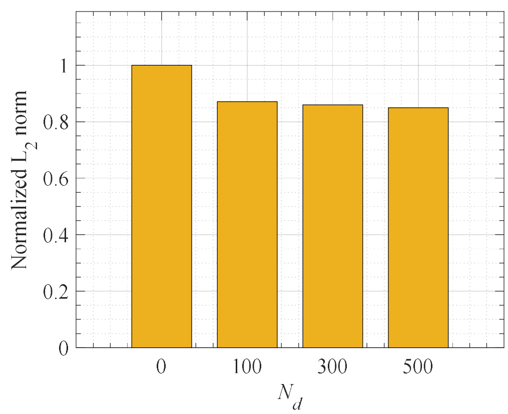

Fig. 8. Normalized L_2-norm values for simulation of Taylor-Green vortex flow with various N_d

4. Tropical Cyclone Boundary-Layer Wind

4.1 Problem formalization

In the boundary layer of a tropical cyclone, the weakly-coupled dynamics and thermodynamics are usually independently examined (e.g., Meng et al. 1995; Kepert 2001; Snaiki and Wu 2017a, b). Accordingly, the wind fields are governed by reduced 3D Navier-Stokes equations:

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v}.\nabla \boldsymbol{v} = -\frac{1}{\rho}\nabla p - \left(2\Omega \sin\varphi\right)\boldsymbol{k}\times\boldsymbol{v} + \boldsymbol{F} \tag{8}$$

where $\boldsymbol{v}$ = wind velocity; Ω = rotation rate of the Earth; φ = Latitude; $\boldsymbol{k}$ = unit vector in the vertical direction; and $\boldsymbol{F}$ = frictional force. The above-mentioned horizontal momentum equations are typically solved with a prescribed Holland pressure expressed as (Holland 1980):

$$p = p_c + \Delta p \exp\left[-\left(r_m / r\right)^B\right] \tag{9}$$

where p_c = central pressure; Δp = central pressure difference; r_m = radius of maximum winds; r = radial distance from the tropical cyclone center; and B is Holland's radial pressure parameter ($B = 1$ in this study). Equation (8) is supplemented by the continuity equation which is expressed for the case of incompressible flow as $\nabla.\boldsymbol{v} = 0$. In a cylindrical coordinate system $\left(r, \theta, z\right)$, Eq. (8) can be expressed as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} + \frac{v}{r}\frac{\partial u}{\partial \theta} + w\frac{\partial u}{\partial z} - \frac{v^2}{r} - \left(2\Omega\sin\varphi\right)v = -\frac{1}{\rho}\frac{\partial p}{\partial r} + K_m\left[\nabla^2 u - \frac{1}{r^2}\left(u + 2\frac{\partial v}{\partial \theta}\right)\right] \tag{10a}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial r} + \frac{v}{r}\frac{\partial v}{\partial \theta} + w\frac{\partial v}{\partial z} + \frac{uv}{r} + \left(2\Omega\sin\varphi\right)u = K_m\left[\nabla^2 v - \frac{1}{r^2}\left(v - 2\frac{\partial u}{\partial \theta}\right)\right] \tag{10b}$$

where $\left(u, v, w\right)$ = wind velocity components; and K_m = eddy viscosity. While a constant eddy viscosity is employed here, a more accurate consideration (e.g., spatially varying values) can be readily implemented in the simulation when data are available. Using the scale analysis (e.g., Snaiki and Wu 2017a; 2017b; Fang et al. 2018), Eqs. (10a) and (10b) can be further simplified as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} + \frac{v}{r}\frac{\partial u}{\partial \theta} + w\frac{\partial u}{\partial z} - \frac{v^2}{r} - \left(2\Omega\sin\varphi\right)v = -\frac{1}{\rho}\frac{\partial p}{\partial r} + K_m\frac{\partial^2 u}{\partial z^2} \tag{11a}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial r} + \frac{v}{r}\frac{\partial v}{\partial \theta} + w\frac{\partial v}{\partial z} + \frac{uv}{r} + \left(2\Omega\sin\varphi\right)u = K_m\frac{\partial^2 v}{\partial z^2} \tag{11b}$$

In addition, the continuity equation can be expressed in the cylindrical coordinates as:

$$\frac{1}{r}\frac{\partial ru}{\partial r} + \frac{1}{r}\frac{\partial v}{\partial \theta} + \frac{\partial w}{\partial z} = 0 \tag{11c}$$

To solve for the solutions of the wind field components, the boundary conditions at the upper atmosphere and near the surface are required. At the surface level, the widely-used bulk formulation with drag coefficient C_D is utilized as (e.g., Kepert and Wang 2001):

$$K_m \left.\frac{\partial u}{\partial z}\right|_{z=0} = C_D u\sqrt{u^2+v^2} \tag{12a}$$

$$K_m \left.\frac{\partial v}{\partial z}\right|_{z=0} = C_D v\sqrt{u^2+v^2} \tag{12b}$$

At the top of the boundary-layer, it is assumed that the frictional wind components is negligible. Hence, the gradient wind balance is established as:

$$\frac{v_g^2}{r} + \left(\left(2\Omega\sin\varphi\right) + \frac{c\sin(\theta-\theta_0)}{r}\right)v_g - \frac{1}{\rho}\frac{\partial p}{\partial r} = 0 \tag{13}$$

where v_g = gradient wind speed; c = tropical cyclone translational wind speed; and θ_0 = approach angle (counterclockwise positive from the East). Therefore, the gradient wind speed can be analytically determined as:

$$v_g = \frac{-c\sin(\theta-\theta_0) - \left(2\Omega\sin\varphi\right)r}{2} + \sqrt{\left(\frac{c\sin(\theta-\theta_0) + \left(2\Omega\sin\varphi\right)r}{2}\right)^2 + \frac{r}{\rho}\frac{\partial p}{\partial r}} \tag{14}$$

In this study, the gradient wind level is set to occur at a height of $z_g = 1500m$. According to the governing equations, the tropical cyclone boundary-layer wind simulation requires both the spatial coordinates (r,θ,z) and several parameters that essentially characterize the storm structure (denote by $\alpha = [\Delta p; r_m; c; \theta_0; \varphi; z_0]$ here) as the inputs. The central pressure difference Δp, radius of maximum winds r_m, translational speed c, approach angle θ_0 and hurricane center location φ can be readily retrieved from the National Hurricane Center, and the surface roughness z_0 (to obtain the drag coefficient through $C_D = \kappa^2 / \left[\ln\left(10/z_0\right)\right]^2$ and κ = von Karman constant) can be obtained using the available database on Land Use/Land Cover (e.g., Hansen 1993; Wieringa 1992; 1993). It should be noted that the unsteady term related to the gradient wind can be expressed as $-c.\nabla v_g$, while the unsteady term related to the frictional wind can be ignored as it is significantly smaller than turbulence viscosity and inertia terms (e.g., Meng et al. 1995).

4.2 Knowledge-enhanced deep learning for 3D wind simulation

The tropical cyclone boundary-layer winds are approximated by a knowledge-enhanced deep learning with a hyperbolic tangent activation function and a network architecture of 9-100-100-100-100-100-100-100-3, where the nine input parameters are r, θ, z, Δp, r_m, c, θ_0, φ and z_0. A simplified representation of the deep neural network architecture enhanced by an integration of both rationalism-based and empiricism-based knowledge is presented in Fig. 9.

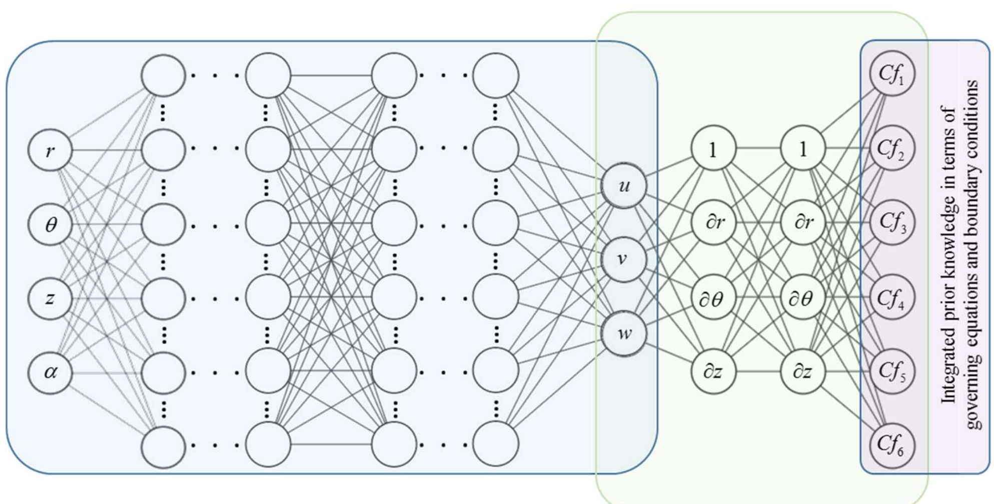

Fig. 9. Knowledge-enhanced deep network architecture for tropical cyclones

The governing equation-based cost functions Cf_1, Cf_2, and Cf_3 are given as:

$$Cf_1 = u\frac{\partial u}{\partial r} + \frac{v}{r}\frac{\partial u}{\partial \theta} + w\frac{\partial u}{\partial z} - \frac{v^2}{r} - (2\Omega\sin\varphi)v + \frac{1}{\rho}\frac{\partial p}{\partial r} - K_m\frac{\partial^2 u}{\partial z^2} \tag{15a}$$

$$Cf_2 = u\frac{\partial v}{\partial r} + \frac{v}{r}\frac{\partial v}{\partial \theta} + w\frac{\partial v}{\partial z} + \frac{uv}{r} + (2\Omega\sin\varphi)u - K_m\frac{\partial^2 v}{\partial z^2} \tag{15b}$$

$$Cf_3 = \frac{1}{r}\frac{\partial ru}{\partial r} + \frac{1}{r}\frac{\partial v}{\partial \theta} + \frac{\partial w}{\partial z} \tag{15c}$$

The boundary-condition based cost functions Cf_4, Cf_5 and Cf_6 are presented as:

$$Cf_4 = K_m \left.\frac{\partial u}{\partial z}\right|_{z=0} - C_D u\sqrt{u^2+v^2}\Big|_{z=0} \tag{16a}$$

$$Cf_5 = K_m \left.\frac{\partial v}{\partial z}\right|_{z=0} - C_D v\sqrt{u^2+v^2}\Big|_{z=0} \tag{16b}$$

$$Cf_6 = v\big|_{z=z_g} - \left[\frac{-c\sin(\theta-\theta_0)-(2\Omega\sin\varphi)r}{2} + \sqrt{\left(\frac{c\sin(\theta-\theta_0)+(2\Omega\sin\varphi)r}{2}\right)^2 + \frac{r}{\rho}\frac{\partial p}{\partial r}}\right]_{z=z_g} \tag{16c}$$

The total cost function is then expressed as:

$$\begin{aligned} L = & \frac{1}{N_d}\sum_{j=1}^{N_d}\left|Cf_0(r_d^j,\theta_d^j,z_d^j,\alpha_d^j)\right|^2 + \frac{1}{N_f}\sum_{j=1}^{N_f}\left\{\left|Cf_1(r_f^j,\theta_f^j,z_f^j,\alpha_f^j)\right|^2 + \left|Cf_2(r_f^j,\theta_f^j,z_f^j,\alpha_f^j)\right|^2 + \left|Cf_3(r_f^j,\theta_f^j,z_f^j,\alpha_f^j)\right|^2\right\} \\ & + \frac{1}{N_{bs}}\sum_{j=1}^{N_{bs}}\left\{\left|Cf_4(r_{bs}^j,\theta_{bs}^j,z_{bs}^j,\alpha_{bs}^j)\right|^2 + \left|Cf_5(r_{bs}^j,\theta_{bs}^j,z_{bs}^j,\alpha_{bs}^j)\right|^2\right\} + \frac{1}{N_{bt}}\sum_{j=1}^{N_{bt}}\left|Cf_6(r_{bt}^j,\theta_{bt}^j,z_{bt}^j,\alpha_{bt}^j)\right|^2 \end{aligned} \tag{17}$$

where the additional cost function $Cf_0 = (u',v') - (u,v)\big|_{r_d,\theta_d,z_d,\alpha_d}$ includes all the training data from field-measurements or numerical simulations; N_{bs} and N_{bt} refer to the sampled points from surface and top boundary conditions, respectively. In this study, the training datasets contributing to the cost function Cf_0 are obtained based on a tropical cyclone wind model recently developed by Snaiki and Wu (2019), where the accurate and efficient considerations of supergradient winds are highlighted. An alternative is to employ the Weather Research Forecasting model, where the expensive fully-order Navier Stokes equations are numerically solved. While the Xavier's normal initialization algorithm (Glorot and Bengio 2010) can be adopted to initialize the network, a more efficient way (to accelerate convergence) utilized here is based on the initial weights and biases from a trained neural network of a linear tropical cyclone wind model (Snaiki and Wu 2017a). The numbers of randomly distributed spatial points within a region of $0 \le \theta \le 360°$, $0 < r \le 250km$, and

$0 \le z \le 1500\,m$ are $N_d = 10{,}000$, $N_f = 100{,}000$, $N_{bs} = 5000$ and $N_{bt} = 5000$. The L-BFGS-B optimization algorithm is selected to minimize the total loss function (Liu and Nocedal 1989; Byrd et al. 1995).

4.3 Validation and application

4.3.1 Model validation

The trained knowledge-enhanced deep network for simulation of tropical cyclone boundary-layer winds is validated based on historically recorded data of Ivan (2004), Emily (2005), Earl (2010) and Irene (2011) hurricanes. The historical data are derived from the Hurricane Research Division's H*Wind snapshots of the National Oceanic and Atmospheric Administration (Powell et al. 1998). It should be noted that the observations represent the 1-min surface wind speeds (m/s) for the marine exposure. The comparison of the four hurricanes between the observed and simulated results were carried out on the 0000 UTC 15 SEP 2004, 0929 UTC 19 JUL 2005, 1630 UTC 02 SEP 2010, and 0130 UTC 25 AUG 2011, respectively, as shown in Fig. 10. In general, good agreement between the observations and simulations were obtained. While the shapes of modeled and observed wind fields present similar features, some discrepancies can be noted due mainly to the use of idealized pressure profile (Holland's pressure). Furthermore, other environmental factors such as wind shear may also alter the spatial distribution of wind field, but are not considered in the current simulations. The observed maximum wind speeds at the radius of maximum winds are 61.7 m/s at $r_m = 42.5\,km$, 41.7 m/s at $r_m = 61.0\,km$, 56.6 m/s at $r_m = 30.0\,km$ and 47.9 m/s at $r_m = 22.2\,km$ for hurricanes Ivan, Emily, Earl and Irene, respectively. The corresponding simulated wind speeds are 61 m/s, 43.7 m/s, 55.5 m/s and 46.3 m/s, respectively.

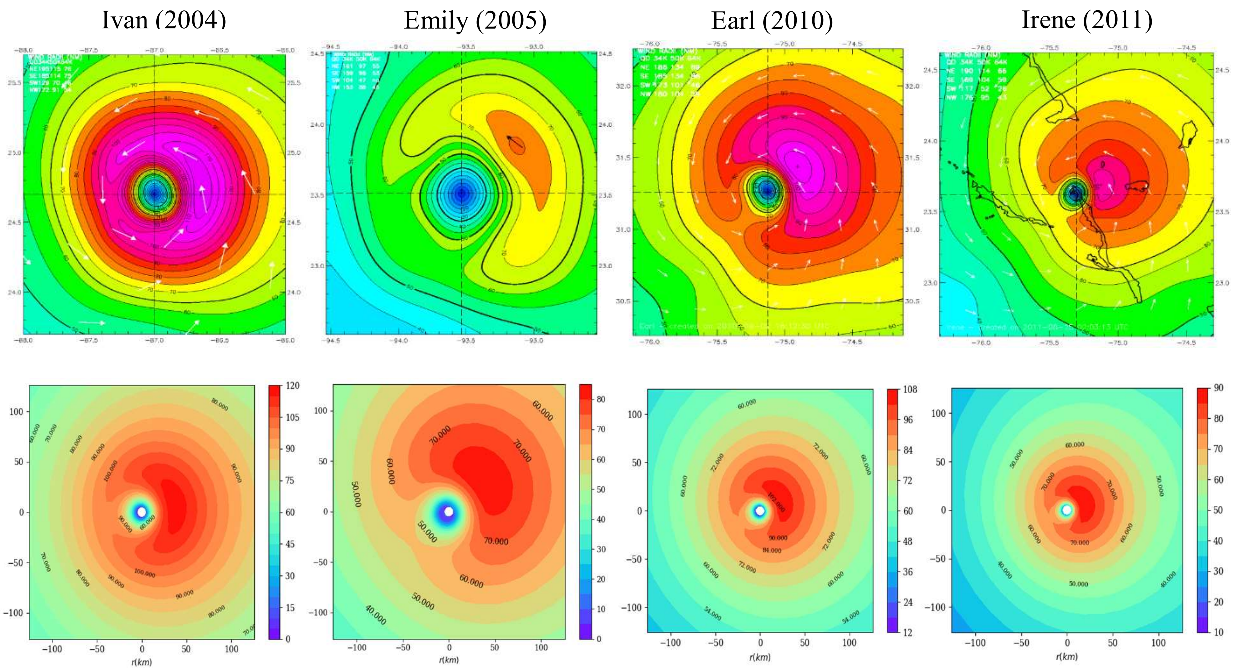

Fig. 10. Comparison between observed (top) and simulated (bottom) surface wind fields of hurricanes Ivan (2004), Emily (2005), Earl (2010) and Irene (2011) [Note: wind speeds are provided in knots]

4.3.2 Model application

Once the developed knowledge-enhanced deep learning is trained and validated, it can be applied to efficiently simulate tropical cyclone boundary-layer winds of an arbitrary scenario. In this study, three tropical cyclone scenarios corresponding to storm parameters listed in Table 1 are investigated.

Table 1. Storm parameters for wind field simulation

Parameter	Δp (hpa)	r_m (km)	c (m/s)	θ_0 (°)	φ (°)	z_0 (m)
scenario 1	100	50	8	70	30	0.001
scenario 2	70	50	4	90	30	0.01
scenario 3	50	50	3	110	30	0.1

Figure 11 depicts the 3D shaded surfaces of simulated wind speed along with the contours of simulated vertical wind profile at the East location (relative to the approach angle). The simulation results indicate that the height of maximum wind decreases with the wind speed. Furthermore, it is noted that an increase of surface roughness leads to a rapid decrease of wind speed near from the ground surface and an increase of central pressure difference results in an increase of wind speed. The supergradient winds, commonly observed in the tropical cyclone boundary layer, are also captured through the developed knowledge-enhanced deep learning since the contribution of horizontal advection, vertical advection and vertical diffusion is considered through the governing equations [i.e., Eqs. (11a) and (11b)].

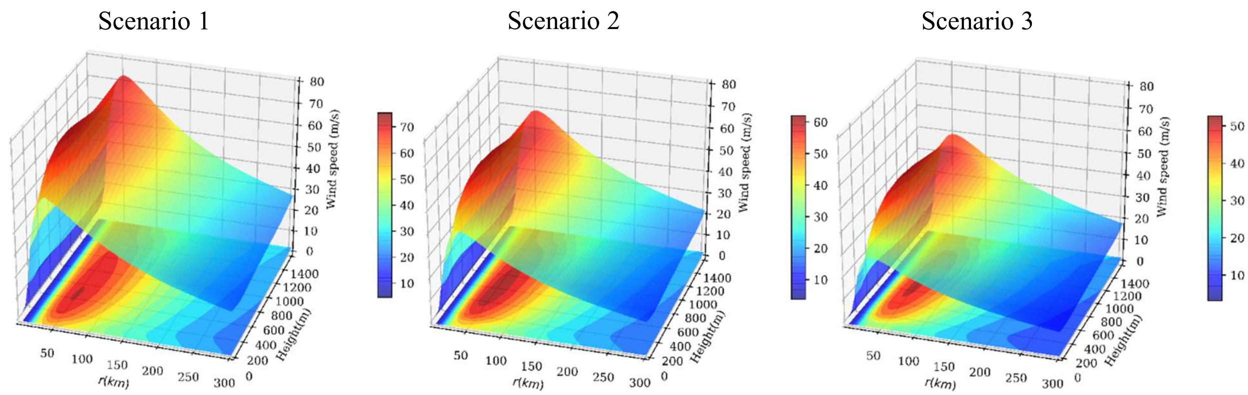

Fig. 11. Three-dimensional shaded surfaces of simulated wind speed and contours of simulated vertical wind profile

Figure 12 further provides the spatial distribution of simulated wind speed at several heights, namely 10m, 500m, 1000m and 1500m. The tropical cyclone asymmetry presented in Fig. 12 is mainly due to the storm translation. The consideration of environmental wind shear (through parametric representations) and spatially varying surface roughness (through Land Use/Land Cover maps) in the developed knowledge-enhanced deep learning may further contribute to the asymmetry of tropical cyclone boundary-layer winds.

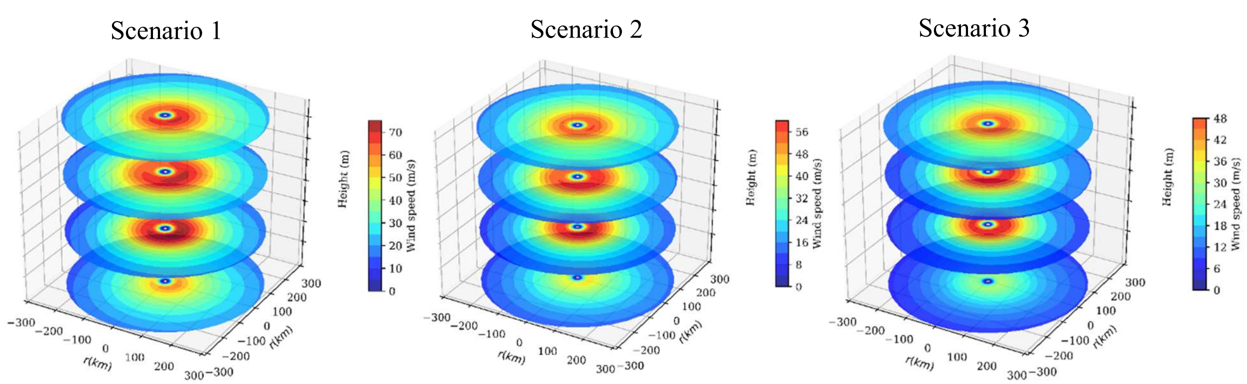

Fig. 12. Spatial distribution of simulated wind speed at 10m, 500m, 1000m and 1500m

To investigate the noise effects on simulation accuracy, all the training data of $N_d = 10,000$ were corrupted with 1%, 3% and 5% uncorrelated Gaussian noise, respectively, for the simulation scenario of $\Delta p = 80hpa$, $r_m = 50km$, $c = 8m/s$, $\theta_0 = 100°$, $\varphi = 30°$ and $z_0 = 0.0001m$. The relative L_2-norm was computed for these three noise cases with respect to the noise-free simulation, and the obtained values are 0.0055, 0.0071 and 0.0093 for the 1%, 3% and 5% uncorrelated Gaussian noise, respectively. This results indicate that a moderate noise in the field-measurement data may not substantially deteriorate the simulation results based on the developed knowledge-enhanced deep network.

While the training of the knowledge-enhanced deep learning discussed here can take on a regular personal computer (CPU @ 3.20 GHz) up to 500,000 seconds, the trained model can rapidly predict the entire wind field within an arbitrary tropical cyclone boundary layer. The execution time of wind prediction depends primarily on the selected spatial resolution, as listed in Table 2.

Table 2. Execution time corresponding to various spatial resolution (CPU @ 3.20 GHz)

dr (km)	$d\theta$ (°)	dz (m)	Wall-clock time (s)
10	10	100	0.58
10	10	50	1.04
5	10	50	1.99
5	10	10	9.5

5. Concluding remarks

In the applications of machine learning (e.g., deep neural network) to engineering problems, it is vital to reduce the demand of big datasets to small datasets considering the high-cost of data generation (using numerical/experimental/field-measurement approach). To this end, the first-step effort is to develop methodologies that leverage rationalism-/empiricism-based knowledge to enhance the purely data-driven approach. Then, the second-step effort is to establish theories as a guide to numerical (e.g., computational-fluid-dynamics), experimental (e.g., wind-tunnel) or field-measurement design for generation of required small datasets for the enhanced machine learning. As preliminary research work of the first-step effort, a knowledge-enhanced deep learning has been proposed in this study to effectively fuse machine-readable prior knowledge in terms of both physics-based equations and semi-empirical formulas governing the wind field inside tropical cyclone boundary layer. The developed knowledge-enhanced deep learning can provide the 3D boundary-layer wind field with high computational efficiency and accuracy for an arbitrary tropical cyclone using the standard storm parameters (i.e., central pressure difference, radius of maximum winds, translational speed, approach angle, latitude of hurricane center and surface roughness), hence, it can be readily utilized in conjunction with the risk analysis framework of tropical cyclone hazards and effectively implemented as part of early system warning. In addition to the refinement of training data and network topology, advances in both scientific knowledge (e.g., proper coupling of tropical cyclone dynamics and thermodynamics) and engineering knowledge (e.g., proper parametrization of pressure, wind shear and friction force) will also contribute to a better performance of the developed knowledge-enhanced deep learning for simulation of tropical cyclone boundary-layer winds.

Acknowledgements

The support for this project provided by the NSF Grant # CMMI 15-37431 is gratefully acknowledged.

Appendix A. A simplified Nonlinear Hurricane Wind Model

To solve Eq. (8), the decomposition method is utilized where the wind velocity ($\boldsymbol{v}$) is expressed as the summation of the gradient wind ($\boldsymbol{v}_g$) and the frictional one ($\boldsymbol{v}'$). While the solution of the gradient wind speed is straightforwardly obtained [i.e., Eq. 14], determining the frictional components needs to numerically solve the nonlinear equations. To save computational cost, the nonlinear governing equation are simplified using the scale analysis approach (e.g., Snaiki and Wu 2017a), and only the nonlinear terms corresponding to the vertical advections, namely $w\frac{\partial u'}{\partial z}$ and $w\frac{\partial v'}{\partial z}$ are retained in the analysis. Accordingly, the following system of equations is obtained:

$$w\frac{\partial u'}{\partial z}-\left(2\frac{v_{\theta g}}{r}+\left(2\Omega\sin\varphi\right)\right)v'=K_m\frac{\partial^2 u'}{\partial z^2} \quad \text{(A.1a)}$$

$$w\frac{\partial v'}{\partial z}+\left(\frac{\partial v_{\theta g}}{\partial r}+\frac{v_{\theta g}}{r}+\left(2\Omega\sin\varphi\right)\right)u'=K_m\frac{\partial^2 v'}{\partial z^2} \quad \text{(A.1b)}$$

To obtain the analytical solutions of the governing equations [Eqs. (A.1a), (A.1b)] involving the vertical advection terms (nonlinear terms), a constant value of w is assumed. While this assumption is reasonably valid near and above the supergradient region based on field measurement and results of numerical models (e.g., Kepert and Wang 2001; Vogl 2009), it is obviously not correct near the surface. However, the vertical wind speed at the very low altitudes (near surface) is negligible compared to the one near and above the supergradient region, and hence the vertical advection can be accordingly neglected. After several mathematical manipulations, the solution of the system of equations is determined as:

$$u'=\sqrt{\alpha/\beta}\exp\left[\left(w/2K_m-x/2\right)z\right]\left[D_1\cos\left(yz/2\right)+D_2\sin\left(yz/2\right)\right] \quad \text{(A.2a)}$$

$$v'=\exp\left[\left(w/2K_m-x/2\right)z\right]\left[-D_1\sin\left(yz/2\right)+D_2\cos\left(yz/2\right)\right] \quad \text{(A.2b)}$$

where $\alpha=\frac{1}{2K_m}\xi_g$; $\beta=\frac{1}{2K_m}\xi_{ag}$; $\xi_g={}^{2v_{\theta g}}\!/_{r}+\left(2\Omega\sin\varphi\right)$; $\xi_{ag}={}^{\partial v_{\theta g}}\!/_{\partial r}+{}^{v_{\theta g}}\!/_{r}+\left(2\Omega\sin\varphi\right)$. D_1, D_2, x and y are given as:

$$D_1 = C_d \left|\boldsymbol{v}_s\right| \left\{ \sqrt{\frac{\beta}{\alpha}} \left[\left(\frac{w}{2K_m} - \frac{x}{2} \right) - C_d \frac{\left|\boldsymbol{v}_s\right|}{K_m} \right] v_{rg} - \frac{y}{2} v_{\theta g} \right\} \Big/ K_m \left\{ \frac{y^2}{4} + \left[\left(\frac{w}{2K_m} - \frac{x}{2} \right) - C_d \frac{\left|\boldsymbol{v}_s\right|}{K_m} \right]^2 \right\} \tag{A.3a}$$

$$D_2 = \left\{ \frac{y}{2} D_1 + \frac{C_d \left|\boldsymbol{v}_s\right|}{K_m} v_{\theta g} \right\} \Big/ \left\{ \left(\frac{w}{2K_m} - \frac{x}{2} \right) - C_d \frac{\left|\boldsymbol{v}_s\right|}{K_m} \right\} \tag{A.3b}$$

$$x = \sqrt{ w^2 \big/ 2K_m^2 + \sqrt{ w^4 \big/ 4K_m^4 + 16\alpha\beta } } \tag{A.3c}$$

$$y = \frac{4\sqrt{\alpha\beta}}{x} \tag{A.3d}$$

The constant value of w is determined based on the continuity equation. Since the vertical and horizontal wind speed are mutually dependent, the iteration approach is utilized in the computation.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016, November. Tensorflow: a system for large-scale machine learning. In OSDI (Vol. 16, pp. 265-283).

Atakulreka, A. and Sutivong, D., 2007, December. Avoiding local minima in feedforward neural networks by simultaneous learning. In Australasian Joint Conference on Artificial Intelligence (pp. 100-109). Springer, Berlin, Heidelberg.

Bardenet, R., Brendel, M., Kégl, B. and Sebag, M., 2013, February. Collaborative hyperparameter tuning. In International Conference on Machine Learning (pp. 199-207).

Baydin, A.G., Pearlmutter, B.A., Radul, A.A. and Siskind, J.M., 2018. Automatic differentiation in machine learning: a survey. Journal of Marchine Learning Research, 18, pp.1-43.

Beidokhti, R.S. and Malek, A., 2009. Solving initial-boundary value problems for systems of partial differential equations using neural networks and optimization techniques. Journal of the Franklin Institute, 346(9), pp.898-913.

Brachet, M.E., Meiron, D.I., Orszag, S.A., Nickel, B.G., Morf, R.H. and Frisch, U., 1983. Small-scale structure of the Taylor–Green vortex. Journal of Fluid Mechanics, 130, pp.411-452.

Burgers, J.M., 1948. A mathematical model illustrating the theory of turbulence. In Advances in applied mechanics (Vol. 1, pp. 171-199). Elsevier.

Byrd, R.H., Lu, P., Nocedal, J. and Zhu, C., 1995. A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing, 16(5), pp.1190-1208.

Canuto, C., Hussaini, M.Y., Quarteroni, A. and Zang, T.A., 2007. Spectral methods: evolution to complex geometries and applications to fluid dynamics. Springer Science & Business Media.

Carrier, G.F., Hammond, A.L. and George, O.D., 1971. A model of the mature hurricane. Journal of Fluid Mechanics, 47(1), pp.145-170.

Constantinescu, R., Lazarescu, V. and Tahboub, R., 2008. Geometrical form recognition using “one-step-secant” algorithm in case of neural network. UPB Sci. Bull., Series C, 70(2).

Czajkowski, J., Simmons, K. and Sutter, D., 2011. An analysis of coastal and inland fatalities in landfalling US hurricanes. Natural hazards, 59(3), pp.1513-1531.

Driscoll, T.A., Hale, N. and Trefethen, L.N., 2014. Chebfun guide.

Fang, G., Zhao, L., Cao, S., Ge, Y. and Pang, W., 2018. A novel analytical model for wind field simulation under typhoon boundary layer considering multi-field correlation and height-dependency. Journal of Wind Engineering and Industrial Aerodynamics, 175, pp.77-89.

Ghosh, J. and Shin, Y., 1992. Efficient higher-order neural networks for classification and function approximation. International Journal of Neural Systems, 3(04), pp.323-350.

Glorot, X. and Bengio, Y., 2010, March. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256).

Hansen, F.V., 1993. Surface roughness lengths (No. ARL-TR-61). ARMY RESEARCH LAB WHITE SANDS MISSILE RANGE NM.

Haykin, S., 1994. Neural networks: a comprehensive foundation. Prentice Hall PTR.

Holland, G.J., 1980. An analytic model of the wind and pressure profiles in hurricanes. Monthly weather review, 108(8), pp.1212-1218.

Hornik, K., Stinchcombe, M. and White, H., 1989. Multilayer feedforward networks are universal approximators. Neural networks, 2(5), pp.359-366.

Huang, W.F. and Xu, Y.L., 2012. A refined model for typhoon wind field simulation in boundary layer. Advances in Structural Engineering, 15(1), pp.77-89.

Huang, W.F. and Xu, Y.L., 2013. Prediction of typhoon design wind speed and profile over complex terrain. Structural Engineering and Mechanics, 45(1), pp.1-18.

Kepert, J., 2001. The dynamics of boundary layer jets within the tropical cyclone core. Part I: Linear theory. Journal of the Atmospheric Sciences, 58(17), pp.2469-2484.

Kepert, J. and Wang, Y., 2001. The dynamics of boundary layer jets within the tropical cyclone core. Part II: Nonlinear enhancement. Journal of the atmospheric sciences, 58(17), pp.2485-2501.

Kepert, J.D., 2010a. Slab‐ and height‐resolving models of the tropical cyclone boundary layer. Part I: Comparing the simulations. Quarterly Journal of the Royal Meteorological Society, 136(652), pp.1686-1699.

Kepert, J.D., 2010b. Slab‐ and height‐resolving models of the tropical cyclone boundary layer. Part II: Why the simulations differ. Quarterly Journal of the Royal Meteorological Society, 136(652), pp.1700-1711.

Khare, S. P., A. Bonazzi, N. West, E. Bellone, and S. Jewson. "On the modelling of over‐ocean hurricane surface winds and their uncertainty." Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography 135, no. 642 (2009): 1350-1365.

Kutz, J.N., 2017. Deep learning in fluid dynamics. Journal of Fluid Mechanics, 814, pp.1-4.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), pp.541-551.

LeCun, Y.A., Bottou, L., Orr, G.B. and Müller, K.R., 2012. Efficient backprop. In Neural networks: Tricks of the trade (pp. 9-48). Springer, Berlin, Heidelberg.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), p.436.

Liu, D.C. and Nocedal, J., 1989. On the limited memory BFGS method for large scale optimization. Mathematical programming, 45(1-3), pp.503-528.

McCulloch, W.S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), pp.115-133.

Meng, Y., Matsui, M. and Hibi, K., 1995. An analytical model for simulation of the wind field in a typhoon boundary layer. Journal of Wind Engineering and Industrial Aerodynamics, 56(2-3), pp.291-310.

Pielke Jr, R.A., Gratz, J., Landsea, C.W., Collins, D., Saunders, M.A. and Musulin, R., 2008. Normalized hurricane damage in the United States: 1900–2005. Natural Hazards Review, 9(1), pp.29-42.

Powell, M.D., Houston, S.H., Amat, L.R. and Morisseau-Leroy, N., 1998. The HRD real-time hurricane wind analysis system. Journal of Wind Engineering and Industrial Aerodynamics, 77, pp.53-64.

Powell, M., Soukup, G., Cocke, S., Gulati, S., Morisseau-Leroy, N., Hamid, S., Dorst, N. and Axe, L., 2005. State of Florida hurricane loss projection model: Atmospheric science component. Journal of wind engineering and industrial aerodynamics, 93(8), pp.651-674.

Psichogios, D.C. and Ungar, L.H., 1992. A hybrid neural network ‐ first principles approach to process modeling. AIChE Journal, 38(10), pp.1499-1511.

Raissi, M., Perdikaris, P. and Karniadakis, G.E., 2017a. Physics Informed Deep Learning (Part I): Data-driven solutions of nonlinear partial differential equations. arXiv preprint arXiv:1711.10561.

Raissi, M., Perdikaris, P. and Karniadakis, G.E., 2017b. Physics informed deep learning (Part II): data-driven discovery of nonlinear partial differential equations. arXiv preprint arXiv:1711.10566.

Rappaport, E.N., 2014. Fatalities in the United States from Atlantic tropical cyclones: New data and interpretation. Bulletin of the American Meteorological Society, 95(3), pp.341-346.

Rosenthal, S.L., 1962. A theoretical analysis of the field of motion in the hurricane boundary layer.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986. Learning representations by back-propagating errors. Nature, 323(6088), p.533.

Shapiro, L.J., 1983. The asymmetric boundary layer flow under a translating hurricane. Journal of the Atmospheric Sciences, 40(8), pp.1984-1998.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural networks, 61, pp.85-117.

Smith, R.K. and Vogl, S., 2008. A simple model of the hurricane boundary layer revisited. Quarterly Journal of the Royal Meteorological Society, 134(631), pp.337-351.

Snaiki, R. and Wu, T., 2017a. A linear height-resolving wind field model for tropical cyclone boundary layer. Journal of Wind Engineering and Industrial Aerodynamics, 171, pp.248-260.

Snaiki, R. and Wu, T., 2017b. Modeling tropical cyclone boundary layer: Height-resolving pressure and wind fields. Journal of Wind Engineering and Industrial Aerodynamics, 170, pp.18-27.

Snaiki, R. and Wu, T., 2018a. An Improved Methodology for Risk Assessment of Tropical Cyclones under Changing Climate. 33rd Conference on Hurricanes and Tropical Meteorology, Ponte Vedra, FL, USA.

Snaiki, R. and Wu, T., 2018b. A semi-empirical model for mean wind velocity profile of landfalling hurricane boundary layers. Journal of Wind Engineering and Industrial Aerodynamics, 180, pp.249-261.

Snaiki, R. and Wu, T., 2019. A Simplified Dynamic System for Estimating Hurricane Supergradient Winds. 15th The International Conference on Wind Engineering (ICWE15), Beijing, China.

Sutton, R.S. and Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.

Swischuk, R., Mainini, L., Peherstorfer, B. and Willcox, K., 2018. Projection-based model reduction: Formulations for physics-based machine learning. Computers & Fluids.

Thornton, C., Hutter, F., Hoos, H.H. and Leyton-Brown, K., 2012. Auto-WEKA: Automated selection and hyper-parameter optimization of classification algorithms. CoRR, abs/1208.3719.

Van Merriënboer, B., Breuleux, O., Bergeron, A. and Lamblin, P., 2018. Automatic differentiation in ML: Where we are and where we should be going. In Advances in neural information processing systems (pp. 8771-8781).

Vickery, P.J. and Twisdale, L.A., 1995. Wind-field and filling models for hurricane wind-speed predictions. Journal of Structural Engineering, 121(11), pp.1700-1709.

Vickery, P.J., Wadhera, D., Powell, M.D. and Chen, Y., 2009. A hurricane boundary layer and wind field model for use in engineering applications. Journal of Applied Meteorology and Climatology, 48(2), pp.381-405.

Vogl, S., 2009. Tropical Cyclone Boundary-Layer Models (Doctoral dissertation, lmu).

Wei, C.C., Peng, P.C., Tsai, C.H. and Huang, C.L., 2018a. Regional Forecasting of Wind Speeds during Typhoon Landfall in Taiwan: A Case Study of Westward-Moving Typhoons. Atmosphere, 9(4), p.141.

Wei, S., Jin, X. and Li, H., 2018b. General solutions for nonlinear differential equations: a deep reinforcement learning approach. arXiv preprint arXiv:1805.07297.

Wieringa, J., 1992. Updating the Davenport roughness classification. Journal of Wind Engineering and Industrial Aerodynamics, 41(1-3), pp.357-368.

Wiernga, J., 1993. Representative roughness parameters for homogeneous terrain. Boundary-Layer Meteorology, 63(4), pp.323-363.

Wu, T. and Kareem, A., 2011. Modeling hysteretic nonlinear behavior of bridge aerodynamics via cellular automata nested neural network. Journal of Wind Engineering and Industrial Aerodynamics, 99(4), pp.378-388.

Yoshizumi, S., 1968. On the asymmetry of wind distribution in the lower layer in typhoon. Journal of the Meteorological Society of Japan. Ser. II, 46(3), pp.153-159.