

# Predicting Particle Acceleration Using Flow Feature Extraction in a Louver Particle Separator

Travis Bowman\*, Cairen Miranda†, and John Palmore Jr. ‡§  
Virginia Tech, Blacksburg, VA, 24061

The goal of this work is to facilitate the design of gas turbine engine particle separators by reducing the computational expense necessary to accurately predict particle motion with machine learning (ML). It is well known that particle ingestion has many detrimental impacts on gas turbine engine operation. Particle ingestion is especially of concern in environments where dust, ash, or ice persist. Accordingly, devices called particle separators are used to remove the particles before they enter the engine. The goal of this work is to improve prediction of particle motion in separators. Ice particles are chosen for this study because of their relevance to civil aviation. The louver particle separator developed by [1] with computational studies is used as a baseline for this work. It has been shown that particle acceleration can be reasonably well-predicted in simple laminar flows using deep feed-forward neural networks [2]. However, there exists a gap for predicting particle acceleration in engineering flows such as the louver geometry of [1]. This is the topic of the current paper. The work builds upon work for laminar flows [2]. In the current work, the complex flow around the louver is decomposed into a laminar shear flow and laminar vortex, and results from [2] are used to analyze the decomposed flows. The accuracy of this decomposition for capturing particle trajectory in the louver is evaluated.

## I. Nomenclature

|                                            |   |                                                |
|--------------------------------------------|---|------------------------------------------------|
| $\mathbf{x}$                               | = | Position                                       |
| $\mathbf{v}$                               | = | Particle velocity                              |
| $\mathbf{v}_r = \mathbf{u} - \mathbf{v}$   | = | Relative velocity                              |
| $\mathbf{u}$                               | = | Gas phase velocity                             |
| $\mathbf{a}$                               | = | Particle acceleration                          |
| $\mathbf{a}_{NN}$                          | = | Neural network-predicted particle acceleration |
| $d$                                        | = | Diameter                                       |
| $F$                                        | = | Force                                          |
| $A$                                        | = | Louver anchor length                           |
| $L$                                        | = | Louver length                                  |
| $St$                                       | = | Stokes number                                  |
| $Re_{in} = \rho_g H u_{in} / \mu_g$        | = | Inlet Reynolds number                          |
| $Re_p = \rho_g  \mathbf{v}_r  d_p / \mu_g$ | = | Particle Reynolds number                       |

### Greek letters

|          |   |                        |
|----------|---|------------------------|
| $\rho$   | = | Density                |
| $\mu$    | = | Dynamic viscosity      |
| $\Phi$   | = | Volume fraction        |
| $\tau_p$ | = | Particle response time |
| $\tau_f$ | = | Flow timescale         |

### Subscripts

\*Graduate Student, Department of Mechanical Engineering

†Graduate Student, Department of Mechanical Engineering

‡Assistant Professor, Department of Mechanical Engineering, and AIAA Member

§Corresponding author: [palmore@vt.edu](mailto:palmore@vt.edu)

p = Particle  
g = Gas  
in = inlet  
D = Drag

## II. Introduction

ICE has always posed a threat to aircraft operation. While much of the existing literature on ice involves the study of accretion, this work focuses on ice that has been ingested into the gas turbine engine. Ice can enter the engine in two ways. Large pieces of ice accretion can break off aircraft surfaces and be ingested. Alternatively, existing ice in the atmosphere such as ice crystals or hailstones can be ingested into the engine. Of the two atmospheric ices, hailstones are of much more concern to safe operation of the engine as they typically range from 0.5 cm to 3.0 cm in diameter [3]. It has been well documented that general particle ingestion can have detrimental effects on the operation of gas turbine engines [4]. Ice particle ingestion can cause consequences ranging from surface wear abrasion to power loss. Ice that passes through the fan stage and enters the core presumably melts in the high pressure region of the compressor [5]. Thus, this increases the humidity of the air in the core and changes the operating conditions from the design point [3]. The compressor is highly sensitive to operating condition changes and can enter surge as a result [5].

This paper addresses the prediction of spherical ice particle motion in particle separators. Inertial particle separators are devices that utilize the momentum of solid particles to separate them from a flow. This type of separator is the most applicable to gas turbine engines because of their ability to remove particles from the flow without substantial pressure loss [1]. There are predominantly two styles of particle separators that have been studied: cyclone and particle deflectors with some form of collector. In practice, particle separators are built into the inlet of the engine [1], as would be most practical for ice particle separation. Figure 1 shows an example of a particle deflector type separator. This particular geometry is known as a louver particle separator because of its similarity to the louvers of a window shutter or blind. When it comes to the design of particle separators, there is still much to be learned. The prediction of the ice particle motion inside of the turbulent air flow is challenging and of critical importance to accurately assess particle separation efficiency. One strategy is through numerical simulation using computational fluid dynamics (CFD); however, such studies are computationally expensive. This work is part of an effort to create ML surrogates for particle motion that can be used to simplify separator design. The initial work presented here focuses on developing models that can accurately recreate particle trajectory.

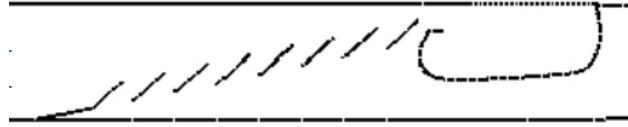


**Fig. 1** Louver particle separator geometry with collector bin [1]. In successful operation, the louvers deflect the particles upwards such that they flow into the collector bin and do not escape. Modified from [1].

### A. Particle Motion Theory

The motion of particles suspended in a fluid can be characterized by the non-dimensional Stokes number,  $St$ , defined by Eq.(1). Physically,  $\tau_p$  can be thought of as the time that it takes for the particle to respond to changes in the gas flow.

$$St = \frac{\tau_p}{\tau_f} \quad (1)$$

$$\tau_p = \frac{\rho_p d_p^2}{18\mu_g} \quad (2)$$

The governing equation for a particle moving through gas, neglecting gravity, can be shown to be:

$$\frac{d\mathbf{v}}{dt} = -f \frac{\mathbf{v} - \mathbf{u}}{\tau_p} \quad (3)$$

where, the function  $f$  introduces a nonlinear drag forcing dependence on the relative velocity. For very small Stokes numbers ( $St \ll 1$ ), the particle motion very closely follows the fluid trajectory, while large particles ( $St \gg 1$ ) act in a more ballistic manner and are not especially driven by the flow stream lines [6]. The goal of this work is to predict particle trajectories using artificial neural networks (ANNs). The approach taken in this study is inspired by the asymptotic solution found by Maxey for the zero-gravity case [7]

$$\mathbf{v} = \mathbf{u} - \tau_p \left( \frac{D\mathbf{u}}{Dt} \right) + O(\tau_p^2) \quad (4)$$

where  $D/Dt$  is the material derivative. When Eq.(4) is substituted back into Eq.(3), the resulting expression gives

$$\frac{d\mathbf{v}}{dt} = \frac{D\mathbf{u}}{Dt} + O(\tau_p) \quad (5)$$

which confirms that the behavior of sufficiently small particles (small  $\tau_p$ ) closely follows that of the streamlines. However, Eq.(5) also hints at a strategy for larger particles. By considering  $\tau_p$  explicitly, higher order terms in the asymptotic expansion can be derived. Rather than manually deriving these terms, they are implicitly derived using machine learning (ML). To achieve this objective, simulations of particles in steady combined laminar flow configurations are performed in the scientific Python environment (*SciPy*), and is discussed in more detail in the Methods section. Particles of various diameters corresponding to Stokes numbers from 0.05 - 500 are considered to capture the influence of  $St$ . These simulations are used to extract relevant particle data to train and validate a neural network for the various flow configurations and combinations thereof.

## B. Machine Learning

The deep feed forward neural network is developed using the open-source ML library *PyTorch*. *PyTorch* is a widely deployed open-source ML framework with a strong research support community. Although the classical application of machine learning is computer vision, current research is showing how ML can be applied to CFD and multiphase flows [8–10]. This work trains the ANN with a well-defined, redundant dataset from combined laminar flows and evaluates how the ML model performs for a particle separator flow using flow feature extraction.

This work extends on the ML methodology presented for particles immersed in laminar canonical flow fields in [2], where the key difference includes a modification of the loss function. Also, for testing the ML model performance with a more realistic flow, this work focuses on a louver-style particle deflector geometry from the results of a computational study performed by [1].

## III. Methods

The scope of particle motion studies conducted assume that the particle volume fraction is small enough such that the dispersed particles do not affect the fluid physics; this is generally referred to as one-way coupled flow. Additionally, the particles considered are assumed to be perfectly spherical ice particles with an isotropic density of  $\rho_g = 917 \text{ kg/m}^3$  [11].

The fundamental principle of this work is the idea that complex flows can be represented as a superposition of simple flows. This idea comes from considering the Taylor expansion of the velocity

$$\mathbf{v} = \mathbf{v}_0 + \Delta x \cdot \nabla \mathbf{v}_0 \quad (6)$$

Over small scales, the influence of complex flow can be represented using only the velocity gradient. The velocity gradient, in turn, can be broken down into a purely rotational and purely straining component. The primary hypothesis of this work is that all complex flows, including flow in a louver particle separator, can be decomposed in this manner. For this work, we consider the decomposition of complex flows using combinations of the forced vortex and irrational shear flows. This velocity field is given by

$$\mathbf{u} = [(e - \omega)x_2, (e + \omega)x_1] \quad (7)$$

Here  $e$  is the shear rate and the rotation rate is  $\omega$ . This work will compare results of particle accelerations from actual louver flows and from superimposed simple flows to evaluate the accuracy of the primary hypothesis.

### A. Neural Network Model

Following [12], a rectangular-like block structure ANN was tested and slightly tuned. The rectangular block structure has the same number of neurons in each HL, and this work refers to the structure as rectangular-like because the last two HL are made smaller such that some form of abstract generalization is forced. This ANN structure proved more feasible with the combined flow particle data from  $\theta = 15^\circ - 360^\circ$ . Specifically, a 7 HL rectangular-like block structure ANN (150 neurons in first five HL and 100 in the last two HL) was used. Rather than using the pure mean squared error (MSE) between the actual and ANN predicted accelerations, a custom loss function was defined to account for the physical nature of acceleration. It is discussed in Section IV.

#### 1. Training Data Generation

The governing particle dynamics equations are shown in Eq.(8) and Eq.(9) where  $\mathbf{x}_p$  and  $m_p$  are the particle position and mass, respectively. The drag force,  $\mathbf{F}_D$ , is given by Eq.(10) where the relative velocity is given by  $\mathbf{v}_r = \mathbf{u} - \mathbf{v}$  and  $A$  is the projected particle area normal to the flow. The drag coefficient,  $c_D$ , is taken to be the Schiller-Naumann relation given in Eq.(11) [13].

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{v} \quad (8)$$

$$m_p \frac{d\mathbf{v}}{dt} = \mathbf{F}_D \quad (9)$$

$$\mathbf{F}_D = \frac{1}{2} c_D \rho_g |\mathbf{v}_r|^2 A \frac{\mathbf{v}_r}{|\mathbf{v}_r|} \quad (10)$$

$$c_D = \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687}) \quad (11)$$

To generate the ANN training data, the governing equations (Eq.(8) and Eq.(9)) of the particle dynamics are solved using numerical integration techniques in the scientific Python computing environment (*SciPy*). The particle initial conditions are taken as  $\mathbf{v}(t_0) = [0, 0]$  and  $\mathbf{x}_p(t_0) = [x_{1,0}, x_{2,0}]$ , where the initial positions  $x_{1,0}$  and  $x_{2,0}$  are randomly selected. However,  $x_{1,0}$  and  $x_{2,0}$  are both restricted to initialize in range  $[-1, 1]$ . Specifically, because canonical flow systems can present stiff systems of differential equations for the particle trajectory, the LSODA method was employed [14]. Some brief testing was performed with stiff ODE solvers in *SciPy* including LSODA, Radau, and backwards differentiation. It was found that LSODA was more computationally efficient for this problem, and hence it was selected.

The velocity fields for the study are synthetically generated using the combined flow field, Eq. (7). The strength of the flow field is measured with second principal invariant of the velocity gradient, defined in Eq.(12).

$$I_2(\nabla \mathbf{u}) = \frac{1}{2} \left( (tr(\nabla \mathbf{u}))^2 - tr(\nabla \mathbf{u}^2) \right) \quad (12)$$

For an incompressible flow, Eq.(12) can be shown to simplify as

$$I_2(\nabla \mathbf{u}) = \frac{1}{2} \frac{\partial u_i}{\partial u_j} \frac{\partial u_j}{\partial u_i} \quad (13)$$

where Einstein summation convention is used. We define  $J(\nabla \mathbf{u}) = \sqrt{2I_2(\nabla \mathbf{u})}$ . This quantity can be thought of as a type of magnitude for the velocity gradient tensor. This definition can also be applied to the symmetric part of the tensor ( $\mathbf{S}$ ) and antisymmetric part of the tensor ( $\mathbf{\Omega}$ ) of the velocity gradient to yield strengths  $S$  and  $\Omega$ , respectively. Not surprisingly,  $\Omega = |\omega|$ ,  $S = e$ , and  $J = \sqrt{S^2 + \Omega^2}$ .

A grid of  $e$  and  $\omega$  values was generated based upon a unit circle idea. Note that  $J = \sqrt{e^2 + \omega^2}$  and define  $\theta = \tan^{-1}(\omega/e)$ , then  $e = J \cos \theta$  and  $\omega = J \sin \theta$ . Fixing,  $J$ , the quantities  $e$  and  $\omega$  are uniquely defined by choosing  $\theta$ . For a given  $J$  value, this work sweeps through  $\theta \in [0^\circ, 360^\circ)$  in  $15^\circ$  increments.

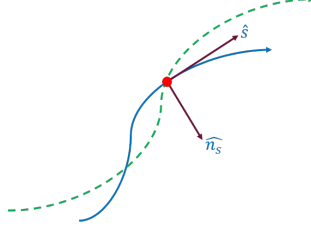


**Fig. 2 Streamline coordinates.** The blue line represents the fluid streamline. The green dashed line represents the particle's path. Vectors are shown for the streamwise and normal unit vectors at the particle's instantaneous position.

Furthermore, to non-dimensionalize the model inputs the governing equations are modified as

$$\frac{d\mathbf{v}^*}{dt^*} = \frac{1}{2} \frac{\rho_g}{\rho_p} \frac{L_{nd} A_p^* c_D^* \|\mathbf{v}_r^*\|^2}{V_p^*} \frac{\mathbf{v}_r^*}{\|\mathbf{v}_r^*\|} \quad (14)$$

$$\frac{d\mathbf{x}_p^*}{dt^*} = \mathbf{v}^* \quad (15)$$

$$c_D^* = \frac{24}{Re_p^*} (1 + 0.15(Re_p^*)^{0.687}) \quad (16)$$

$$Re_p^* = \frac{\mathbf{v}_r^* d_p^*}{\nu^*} \quad (17)$$

where the dynamic viscosity  $\nu = \mu_g / \rho_g$ ,  $(\cdot)^*$  indicates a non-dimensionalized parameter (e.g.  $d_p^* = \frac{d_p}{L_{nd}}$ ). The natural choice for the timescale to non-dimensionalize is the factor,  $\frac{1}{J}$ , and the length scales are non-dimensionalized with  $L_{nd} = 1.0cm$  based on the order of magnitude that hailstones normally exist at [15]. The relevant equations are solved in non-dimensional form using  $L_{nd} = 1.0cm$  and  $\frac{1}{J}$  as the characteristic values.

Five different particle sizes corresponding to  $St = 5 \times [10^{-2}, 10^{-1}, 10^0, 10^1, 10^2]$  are considered in this study. The relevant parameters are extracted for 5,000 particles of each  $St$  at  $t^* = 1.0$  ( $t = t^* J$ ), and 80% of the data are used for training the ANNs and the remaining 20% are used for validation.

The ANN model inputs are given as

$$\frac{d\mathbf{v}^*}{dt^*} = g \left( \mathbf{v}^*, \mathbf{u}^*, d_p, \frac{D\mathbf{u}^*}{Dt^*}, \rho_p, \rho_g, \mu_g, \frac{e}{J}, \frac{\omega}{J} \right) \quad (18)$$

such that the model can use the relative contributions of rotation and strain at the particle location as a means of flow feature extraction for general flows. The training data is preprocessed using a transformation of coordinates to streamline coordinates as suggested in [2], and it is shown in Figure 2. The solid line in Figure 2 represents a fluid streamline and the dashed line represents a particle path. The streamline coordinates are represented by  $\hat{s}-\hat{n}_s$ , where  $\hat{s}$  is tangent to the fluid streamline (i.e. the direction of  $\mathbf{u}$ ) and  $\hat{n}_s$  is the corresponding normal vector.

## B. CFD Solver and Computational Setup

Machine learning data is compared to particle flow analysis in the actual louver. For simulations of the louver geometry in this study, Eulerian-based gas phase transport equations and Lagrangian-based particle tracking equations are utilized for the unique challenges that particle laden flows present [16] [17]. This Euler-Lagrange method uses Direct Numerical Simulation (DNS) to solve the transport equations, which are given by the Navier-Stokes equations. Following Miranda and Palmore [18], the particles are modeled as discrete, perfectly spherical bodies that are governed by the Lagrangian equations defined by Eqs. (8) to (11). The equations used by Miranda and Palmore are simplified such that the forces due to particle collisions are not included because these multiphase simulations fall under the regime of one-way coupling, and gravitational forces are neglected as well.

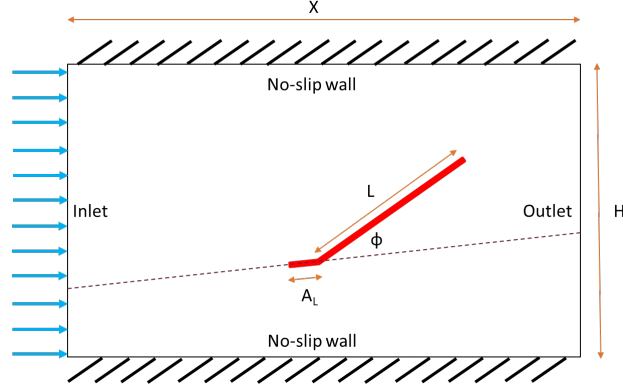


**Fig. 3 Louver particle separator computational domain. The top and bottom of the domain are no-slip walls, the flow into the domain is a velocity inlet where particles are also initialized with the same inlet velocity as the fluid, and the outlet allows particles to escape the domain.**

The inlet is specified as a velocity inlet and the top and bottom of the domain are no-slip walls as shown in Figure 3. For each particle size, the domain size is scaled so the ratio of  $L/d_p$  is not altered. The inlet velocity  $u_{in}$  is then specified to fix  $Re_{in}$ , and the wall boundary conditions are specified since particle separators are designed for either inlet or secondary flow particle separation [1]. Additionally, the louver geometry is taken as a no-slip wall and the outlet boundary condition is a convective outflow that also allows the particles to escape the domain. Particles are initialized with the same velocity as the fluid at the inlet. These simulations are designed to ensure that the volume fraction of particles to fluid,  $\Phi$ , does not exceed  $10^{-6}$ , such that the governing equations reflect a one-way coupled scheme [19]. The single louver, shown in the middle of the domain, is defined by the louver and anchor lengths  $L$  and  $A$ , respectively, and the louver angle  $\phi$ .

#### IV. ANN Setup

The use of an artificial neural network (ANN) requires the selection of a large number of parameters in an *ad hoc* manner. The process of choosing the parameters are typically not demonstrated in the open literature. Here, a major focus of the work is to demonstrate the relative importance of the various strategies in designing a neural network.

##### A. Choice of Loss

From a mathematical perspective, training a neural network consists of selecting a set of parameters that minimizes a certain objective function, called the loss. The loss is usually taken to be some functional applied to the error from the training. The most common choice for the functional is the mean square operation so that the loss is the mean square error (MSE). In [2], the MSE was used initially, however, it was discovered that the accuracy of the neural network prediction was strongly dependent on particle size. In order to equally penalize the weights and biases for accelerations of small and large particles, (small particles will experience much larger accelerations compared to the large particles) a custom loss function is defined for the ANN.

To define the loss, a few definitions are in order. The error (vector) is defined with the difference between actual particle acceleration (predicted from detailed fluid dynamics) and the neural net,

$$\mathbf{e} = \mathbf{a} - \mathbf{a}_{NN} \quad (19)$$

Error can also be measured based on the prediction of the magnitude predicted by the neural net and the direction,

$$E = |\mathbf{a}| - |\mathbf{a}_{NN}| \quad (20)$$

$$\beta = \frac{\mathbf{a} \cdot \mathbf{a}_{NN}}{|\mathbf{a}| |\mathbf{a}_{NN}|} \quad (21)$$

Initially, a loss function defined as  $loss = c_1 * MSE(E) + c_2 * MSE(1 - \cos \beta)$  was used, where  $c_1$  and  $c_2$  are hyperparameters that should be tuned. However, this approach did not work well as it is impossible to know how  $E$

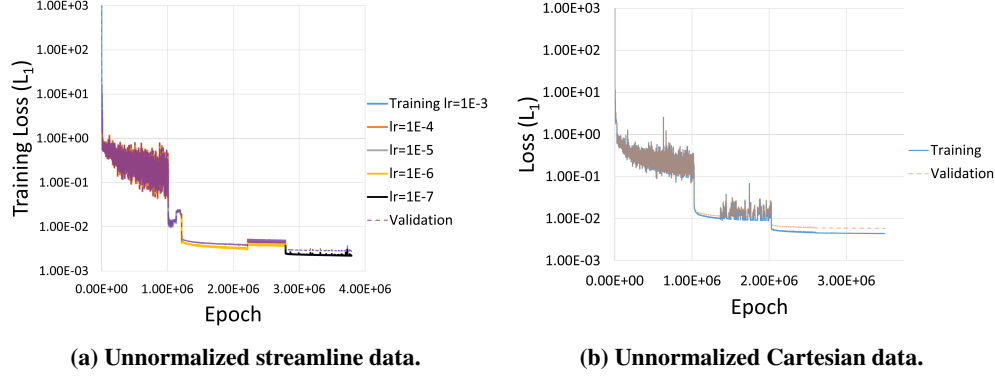


**Fig. 4** Use of  $L_1$  loss function. The  $L_1$  loss function is less sensitive to outliers and allows the ANN to achieve a lower loss. Note that the  $lr$  in b) was iteratively limited similarly to a).

**Table 1** Master ANN Data Preprocessing Effects. Trained with Adam using SELU activation function and custom  $L_1$  loss function. Final training and validation loss after iteratively decreasing  $lr$  to  $1.0E-7$  is shown.

| Coordinate System | Training Loss | Validation Loss |
|-------------------|---------------|-----------------|
| Cartesian         | 4.45E-03      | 5.91E-03        |
| Streamline        | 2.22E-03      | 2.83E-03        |

scales *a priori*. Since the term  $1 - \cos \beta$  is always of order unity, the lack of scaling for  $E$  meant that the neural network trained scaled exclusively on magnitude and not direction. An alternative approach was introduced as

$$\epsilon = \frac{e}{\|a\|} \quad (22)$$

$$\text{loss} = \text{MSE}(\epsilon) \quad (23)$$

This has the advantage of having a natural scaling built into the definition of  $\epsilon$ , however the disadvantage that the relative importance of the magnitude error and directional error cannot be tuned separately.

Another important parameter in training is the learning rate,  $lr$ . It controls sensitivity of the training to variations in the loss. During training, it was found that manually limiting the learning rate was conducive to further convergence. This was surprising based on the fact that the Adam optimizer computes adaptive learning rates of its own for various parameters [20]. As the convergence of the training appeared to level off, the  $lr$  was decreased by a factor of 10, and the training was restarted. This process was repeated until a desired loss was achieved, or no further decrease in the loss can be established through the process.

Through all of the previous studies, the MSE loss function had been used as it is a standard workhorse in neural network research. However, the MSE is known to be sensitive to outlier data. Hence, training is slowed down by this sensitivity, and a loss function is based on an alternative approach. The  $L_1$  method, related to the mean absolute error (MAE), was chosen due to its lower sensitivity to outliers. Figures 4a shows the result of training the network with  $L_1$ . The final  $L_1$  training loss achieved was  $2.2E-3$  as shown in Table 1, which is an order of magnitude lower than the same network trained using the MSE loss. An addition test was done to repeat a result of [2] which demonstrated that preprocessing the data into streamline coordinates trained better than Cartesian coordinates. The neural network was retrained with the Cartesian data set, and the resulting training and validation trends are shown in Figure 4b. As expected, the final training loss was higher, in fact about twice as large at  $4.4E-3$ , than the streamline data test as shown in Table 1. This agrees with and bolsters the results shown in [2]. The validation loss for the Cartesian data test is also proportionally higher than that of the streamline data.

## B. ANN Validation

Using the trained ANN from Figure 4a, the validation data were analyzed in detail as shown in Figure 5. This time, however, it was appropriate to include a plot of the  $L_1$  error vs.  $\theta$  norm as shown in Figure 5d. Much like the validation

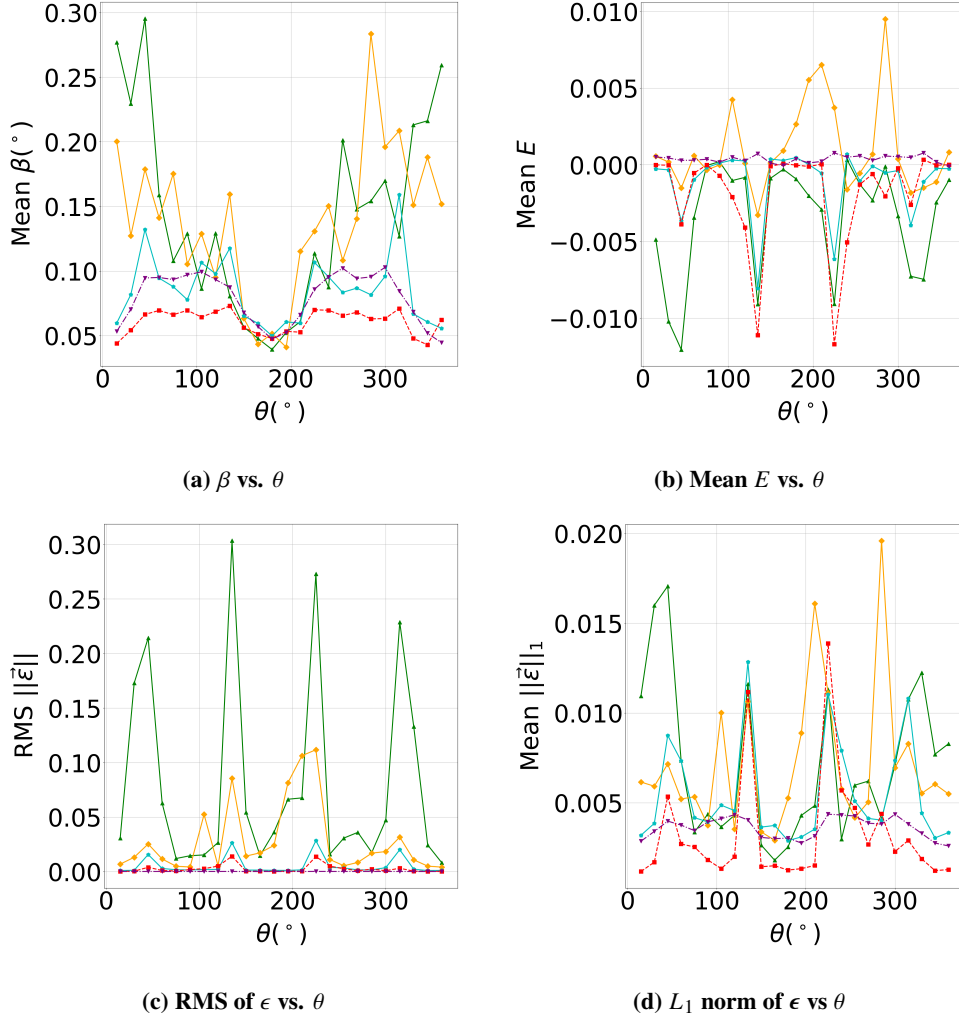


**Fig. 5** Improved ANN combined flow errors for  $St = 0.05$  ( $\blacktriangle$ ),  $0.5$  ( $\blacklozenge$ ),  $5$  ( $\bullet$ ),  $50$  ( $\blacksquare$ ) and  $500$  ( $\blacktriangledown$ ).

results presented in [2], the smallest two  $St$  exhibit some symmetry about  $\theta = 180^\circ$ , but for this ANN all  $St$  directional errors have some amount of symmetry about  $\theta = 180^\circ$ . Additionally, the values of  $\beta$  are much more acceptable than before. The maximum average  $\beta$  for any combination of  $St$  and  $\theta$  does not exceed  $0.3^\circ$ , while previous results for the larger  $St$  had most average  $\beta$  on the order of  $60 - 80^\circ$ . As shown in Figure 5b, the mean absolute value of  $E$  only slightly exceeds 0.01 for three combinations of  $St$  and  $\theta$ . This means, on average, the predicted acceleration magnitude is within 1% of  $\mathbf{a}$ . In fact, most of the values are below 0.005. The main values of  $\theta$  where the peaks occur are around  $45^\circ$ ,  $135^\circ$ ,  $225^\circ$ , and  $315^\circ$ . For each of these values of  $\theta$ ,  $|\omega| = |e|$  and it can therefore be shown that  $\frac{D\mathbf{u}}{Dt} = 0$ . The model likely has difficulty dealing with zeros in the inputs  $\frac{D\mathbf{u}}{Dt}$  as it likely heavily relies on it for all other data points. Similar trends can be observed in Figure 5c, and it should also be noted that these errors are generally being observed to decrease with  $St$ . This is the exact opposite trend that was observed in [2]. In Figure 5d, the dependence of the  $L_1$  norm of  $\epsilon$  on  $St$  appears to be weakened. This could be for two reasons. Since the ANN was trained with this loss function, it may have successfully balanced the loss from each  $St$  data set equally. Alternatively, it could be simply that the  $L_1$  norm is less sensitive to outliers.

Histograms of the directional error  $\beta$  were plotted by  $St$  as shown in Figure 6. The y axis is displayed with a log scale to indicate the largest errors that occurred. For each  $St$ , the maximum error is no more than  $65^\circ$ . The histograms are remarkably similar, which is likely an ode to the custom loss function. For each case, the fullest bin is the  $0^\circ - 1^\circ$ , which in each case holds clearly over  $10^4$  validation particles.

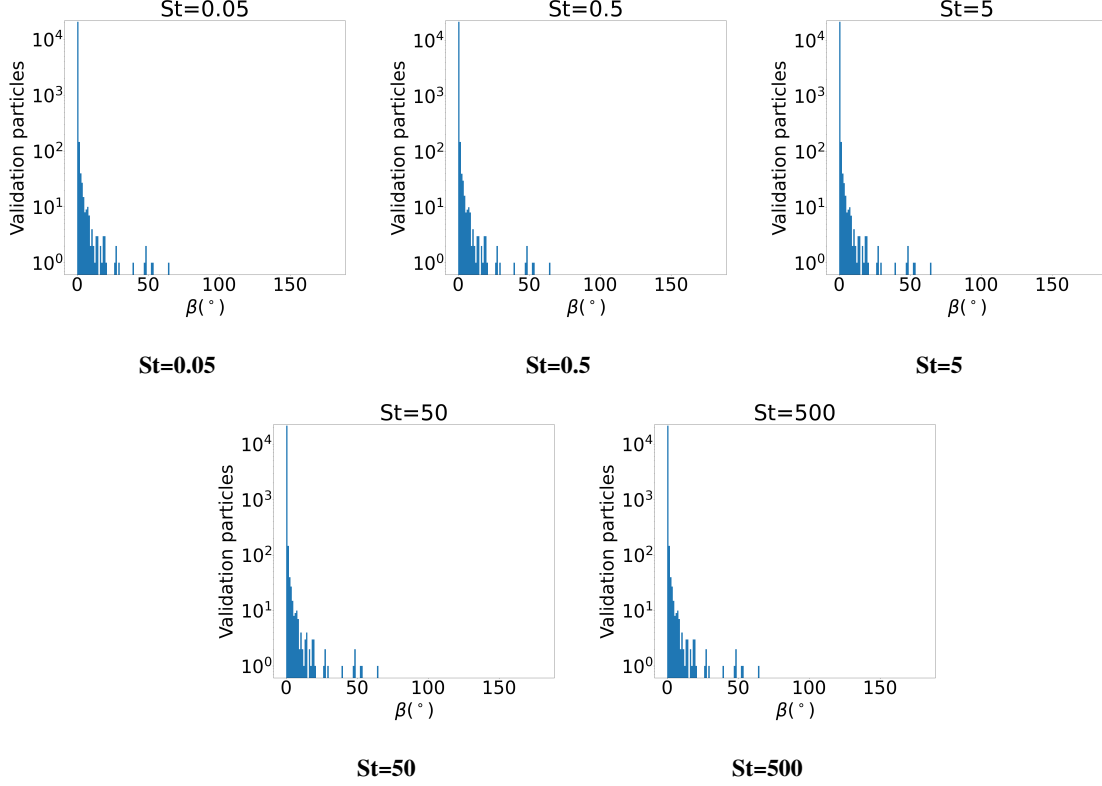


**Fig. 6 Combined flow directional error distribution for the improved ANN. The bin widths are  $1^\circ$**

### 1. Forward Trajectories

The artificial neural network was trained to predict the acceleration of particles based on flow conditions. Although the errors in acceleration are low, the accumulated effect of these errors on the prediction of overall particle trajectories must be tested. For this case, the ANN is used to produce the particle accelerations which are integrated in time, from the particle initial condition. It is compared to the results of solving non-dimensional governing equations given by Eq.(14) and Eq.(15). For the initial test,  $\theta = 90^\circ$  is used to serve as a useful comparison to the ANN trained for the forced vortex in [2].

The results for all  $St$  particles considered in  $\theta = 90^\circ$  are shown in Figure 7. For the zero initial velocity case in Figure 7a, the particle track predicted with the ANN appears to be very short compared to the actual plot of  $\mathbf{x}_p(t)$  for the smaller  $St$  particles. For  $St = 50$  and  $500$ , the governing equations particle tracks are also very short, and will be explored in more detail. For those particles given a random initial velocity shown in Figure 7b, the ANN appears to perform well for six of the ten  $St = 0.05$  particles considered. However, for the other  $St$  considered, the use of  $\mathbf{a}_{NN}$  appears to result in very similar particle location paths as the  $St = 0.05$  particles. To better understand the performance of the Master ANN for the larger  $St$  with  $\mathbf{v}_0 = [0, 0]$ , a single particle track is shown in detail for  $St = 50$  and  $St = 500$  in Figure 8. Clearly, the Master ANN performs well for the  $St = 500$  particle shown while less than half of the particle track is predicted for the  $St = 50$  particle. The superior performance of the ANN for  $St = 0.05$  particles with an initial velocity and the  $St = 500$  particles with zero initial velocity may be an indicator that the ANN is not truly deciphering the dependence of  $\mathbf{a}$  on  $d_p$ . The ANN was trained using data extracted at  $t^* = 1.0$  from the combined canonical flows where the particles were initialized with  $\mathbf{v}_0 = [0, 0]$ . This means that the larger particles shown to the ANN would have a much lower velocity at  $t^* = 1.0$  compared to the smaller particles since large particles experience smaller acceleration in the same flow. As a result, the ANN may have placed unwarranted weight on acceleration's dependence on particle velocity. Thus, the particles initialized with zero velocity in Figure 7a were all determined to have small accelerations and subsequently small displacements. Similarly, the particles initialized with a random velocity were predicted to have large accelerations in the direction of the flow regardless of particle size as shown in Figure 7b. This supports the idea that the Master ANN was over-trained based on  $\mathbf{v}$  and did not decipher the influence of  $d_p$ .

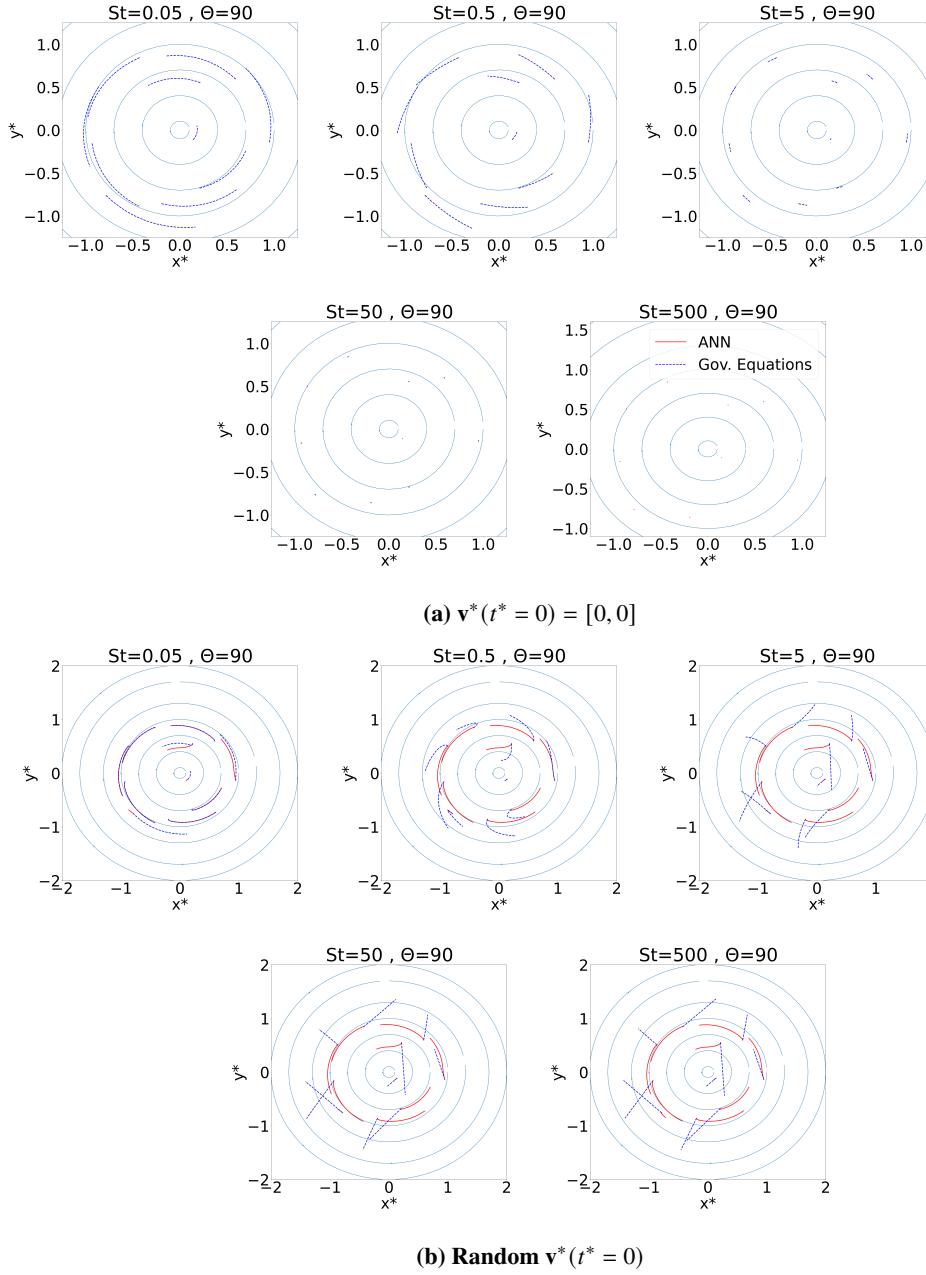


**Fig. 7** Master ANN particle trajectory validation for  $\theta = 90^\circ$ , which corresponds to the counter rotating forced vortex. The vortex is indicated by circular streamlines. The particle trajectories of the governing equation solution and ANN approximation are shown by the dashed blue and solid red lines, respectively. Figure a) shows particles that were initialized with zero velocity, and the particles shown in Figure b) were initialized at the same position but with a random velocity with x and y components in range  $[-1, 1]$ . All plots show the particle trajectory for  $t^* = [0, 1]$ .

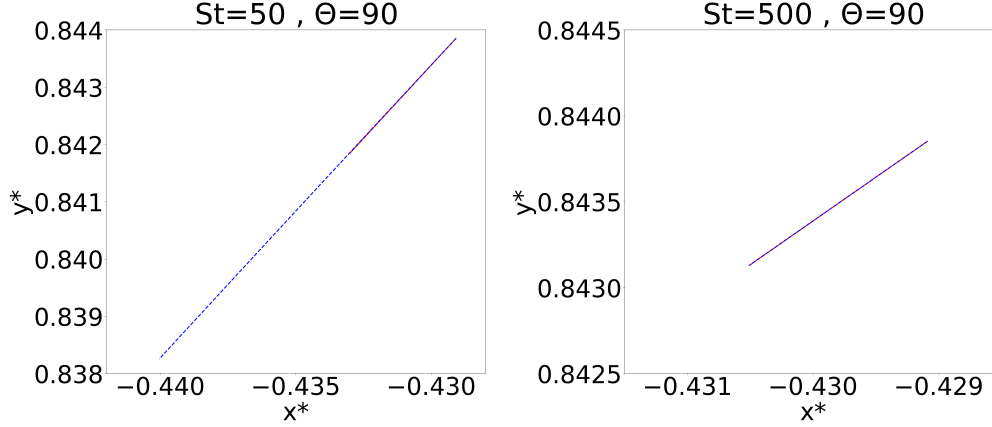


**Fig. 8** Master ANN closeups of large  $St$  single particle tracks from Figure 7. The particle trajectories of the governing equation solution and ANN approximation for  $t^* = [0, 1]$  are shown by the dashed blue and solid red lines, respectively.

## 2. ANN Model Sensitivity

To further validate the idea that the ML model may have over-trained the dependence of  $\mathbf{a}$  on  $\mathbf{v}$ , a simple sensitivity study was conducted. The automatic differentiation package in *PyTorch* was used to compute the sum of the partial derivatives of the two model outputs with respect to each input. Figure 9 shows a box and whisker plot (liers excluded for figure clarity) of this quantity for the entire data set (training and validation) for each model input. Clearly, the model is most sensitive to the stream-wise and normal particle velocity components. While the model does show sensitivity to the particle diameter  $d_p$ , it is less dependent on this than the gas velocity magnitude and normal acceleration component. Additionally, it is not surprising that almost no dependence is shown on the physical properties  $\rho_p$ ,  $\rho_g$ , and  $\mu_g$  because these were constant throughout the data. Interestingly, the model did not develop any significant dependence on the non-dimensionalized contributions of strain and rotation,  $\frac{\epsilon}{J}$  and  $\frac{\omega}{J}$ , respectively. This may suggest that the idea of decomposing complex flows into simple canonical flows is fundamentally flawed. For one-way coupled flow, the particle trajectory may be dominated by local flow information such that the flow gradients do not affect the instantaneous acceleration  $\mathbf{a}$ . Alternatively, because the data set was constructed from particle data at  $t^* = 1.0$ , it is also possible that this data set did not demonstrate the dependence of  $\mathbf{a}$  on  $\frac{\epsilon}{J}$  and  $\frac{\omega}{J}$ . If this is the case, the lack of dependence shown on the tensor invariants is likely due to over-training on  $\mathbf{v}$ .

## C. Neural Network Training for Combined Flows

Figure 10 shows the training results of the ANN as the learning rate (lr) was varied. Specifically, as the training performance appeared to plateau the lr was successively decreased by an order of magnitude. This iterative limiting of the lr appears to have allowed the ML model to better fit the data.

## V. Application to Louver Geometry

The purpose of this project was to investigate the use of machine learning for predicting particle trajectory in a particle separator. The louver particle separator is characterized by Musgrove et al.[1] with the area ratio (AR) and the number of louvers ( $N_L$ ). Physically, the AR represents the ratio of total throat area between louvers to the domain height  $H$  and is defined as  $AR = \frac{\delta N_L}{H}$  [1], where  $\delta$  is the gap between louvers as shown in Figure 11. They found that an  $AR = 1.5$  resulted in the lowest pressure loss of the ARs considered. Additionally, they show that  $N_L = 8$  louvers is optimal for small particle separation efficiency [1]. Removing small particles from the flow is often the most difficult task in particle separator design because sufficiently small particles closely follow the fluid streamlines, as previously discussed in Section II.A. The louver anchors (of length  $A_L$ ) are stacked along a straight line angled to the horizontal shown by  $\gamma$ , which is determined by the space that the louver particle separator must fill [1]. The louvers themselves (of length  $L$ )

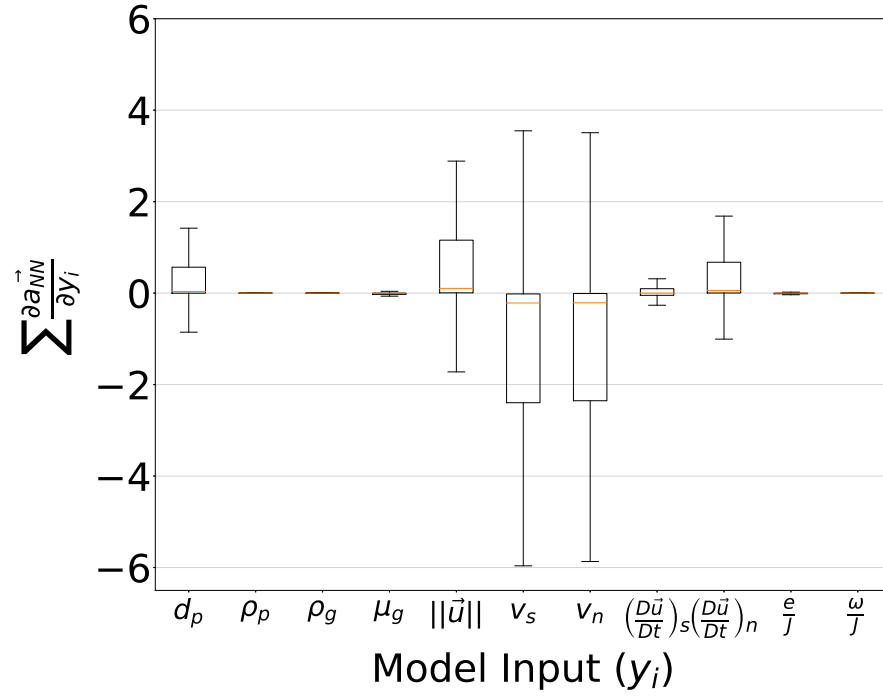


**Fig. 9** Model sensitivity to inputs. The model is clearly more sensitive to the particle velocity inputs.

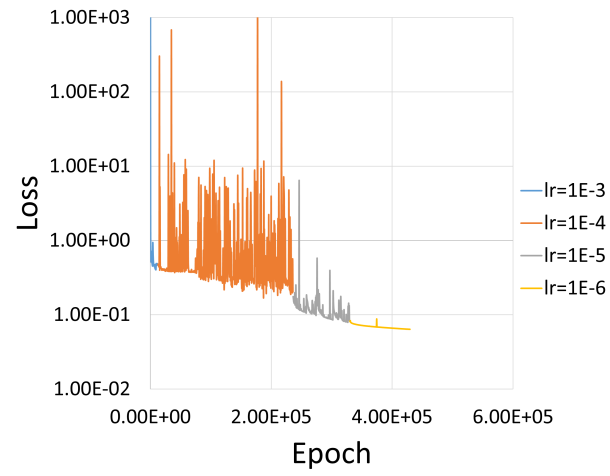


**Fig. 10** ANN training with manually decayed lr.

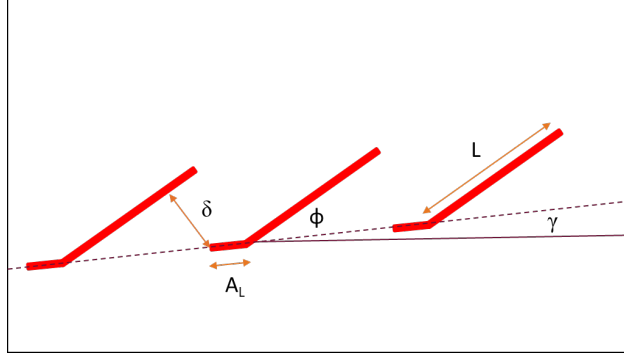


**Fig. 11 Multiple-louver particle separator geometry parameters.**

are set at an angle  $\phi$ , which is defined relative to  $\gamma$  as shown in Figure 11.

The inlet is specified as a velocity inlet and the top and bottom of the domain are no-slip walls as shown in Figure 3. The inlet velocity  $u_{in}$  is specified to fix  $Re_{in}$ , as defined in Eq.(24), where  $H$  is the height of the computational domain. The wall boundary conditions are specified since particle separators are designed for either inlet or secondary flow particle separation for gas turbine engine application [1]. As this setup mimics a pipe-flow case, the  $Re_{in}$  was set to 1000 to ensure that the flow was laminar [21]. This was required since the training data sets were taken exclusively from laminar flows. To ensure a reliable fluid phase solution, the simulation time step  $\Delta t$  was defined in order to limit the Courant-Friedrichs-Lewy (CFL) number to 0.8, as defined in Eq.(25) where  $\Delta x$  is the grid size. Moreover, the particle CFL number  $CFL_p$  defined in Eq.(26) was limited to 0.08, one order of magnitude less than the CFL, to ensure reliable Lagrangian particle tracking.

$$Re_{in} = \frac{\rho_g u_{in} H}{\mu_g} \quad (24)$$

$$CFL = \frac{\Delta t \|\mathbf{u}\|}{\Delta x} \quad (25)$$

$$CFL_p = \frac{\Delta t \|\mathbf{v}\|}{d_p} \quad (26)$$

Additionally, the louver geometry is taken as a no-slip wall, achieved with an immersed boundary method [22], and the outlet boundary condition is a convective outflow that also allows the particles to escape the domain. Particles are initialized at the inlet with the same velocity as  $u_{in}$ . These simulations are designed ensuring that the volume fraction of particles to fluid,  $\Phi$ , does not exceed  $10^{-6}$ , such that the governing equations reflect a one-way coupling scheme [19]. Particle diameters corresponding to Stokes numbers ( $St = \frac{\tau_p}{\tau_f}$ ) of  $St = 5 \times [10^{-2}, 10^{-1}, 10^0]$  are considered. In this case, the particle response time  $\tau_p$  is defined as before in Eq.(2). However, the fluid time scale is defined as  $\tau_f = \frac{\delta}{u_i}$  to reflect the definition of  $St$  given by [23]. The single louver, centered in the domain, is defined by the louver and anchor lengths  $L$  and  $A_L$ , respectively, and the louver angle  $\phi$  which is measured with respect to the louver anchor. Musgrove et al.[1] consider louver angles  $\phi$  from  $30^\circ$  to  $60^\circ$ , and use  $\phi = 30^\circ$  as a baseline comparison. However, note that the optimal geometry presented by [1] varies  $\phi$  from  $30$ - $45^\circ$  along the louver pattern, but this study uses  $\phi = 30^\circ$  since this was the angle used for baseline comparisons. Here,  $L$  is determined implicitly by the relation  $L = \frac{\delta}{\sin(\phi)}$ , after the appropriate value of  $\delta$  had been determined from  $AR = 1.5$  and  $N_L = 8$  in accordance with [1]. While Musgrove et al. vary the louver anchor length to determine an ideal geometry, here the anchor length is taken to be  $A_L = 0.2L$ . In a multi-louver simulation, [1] define the angle of the louver anchor to the horizontal as the angle as a constant angle at which the louvers stack, (i.e. they would stack along the dotted line). For simulating a single louver, this angle is arbitrary, and is chosen to be  $10^\circ$  to maintain similarity with [23].

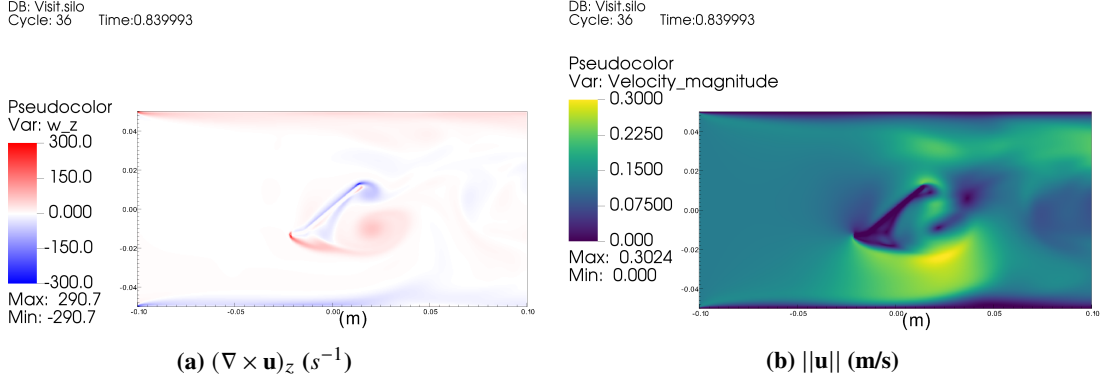


**Fig. 12** Developing flow around the louver geometry.

## VI. Flow Feature Extraction for ANN Evaluation

In order to evaluate the ANN's performance for particles in real flow, it was necessary to compute the non-dimensionalized parameters  $\mathbf{v}^*$ ,  $\mathbf{u}^*$ ,  $\frac{D\mathbf{u}^*}{Dt^*}$ ,  $\frac{d\mathbf{v}^*}{dt^*}$ ,  $S^*$ , and  $\Omega^*$  at each particle position for multiple locations and moments in time from simulation data. It is also required that the times are greater than  $\tau_p$  to decouple the particle statistics from the initial conditions. Specifically, the particle data was extracted every 2000 simulation time-steps ( $2000\Delta t$ ) which corresponds to  $t = 4.8E-2$ ,  $1.6E-1$ , and  $2.5E-1$  s for  $St = 0.05$ ,  $0.5$ , and  $5$ , respectively. These intervals were chosen since they are much greater than  $\tau_p$  for each particle and capture the particles at various points throughout the domain. It was important to record data points at multiple simulation times in order to capture the particle interaction with developing flow and the unsteady wake behind the louver geometry. Recall the definitions of the principal invariants of the strain rate and rotation tensors,  $S$  and  $\Omega$  respectively, given in Section III. As before, time scales are non-dimensionalized with  $\frac{1}{f}$  and length scales are non-dimensionalized by  $L_{nd} = 1.0cm$ . As expected, the flow around the louver was inherently unsteady due to vortex shedding that occurs with blunt bodies immersed in flow. Figure 12a shows the z-component of vorticity for developing flow in the single louver configuration. The vortex region can be clearly seen developing behind the louver. The corresponding velocity magnitude is shown in Figure 12b.

Snapshots of the  $St = 0.05$ ,  $0.5$ , and  $5$  particles are shown after approximately 0.9 seconds of flow time in Figure 13. The  $St$  definition agrees with theory, as the small  $St$  particles appear to trace the flow, even near the louver boundary, while the  $St = 5$  particles appear largely unaffected by the flow streamlines.

### A. ANN Evaluation

As before, the directional error  $\beta$ , magnitude error  $E$ , and normalized error vector magnitude  $\epsilon$  were evaluated as a function of the flow characteristic  $\theta$ . It is important to note that the flow characteristic is calculated as  $\arctan(\frac{\Omega}{S})$  and is therefore limited to the range  $(0^\circ, 90^\circ)$  since  $\Omega$  and  $S$  are positive by definition. Figure 14 shows these three averaged error metrics as a function of  $\theta$  and by  $St$ . From Figure 14a, it is clear that the ANN fails to predict the direction of the acceleration vector. The average value of  $\beta$  is no less than  $45^\circ$  for any value of  $\theta$ . The averaged values of  $E$  and  $\epsilon$  are remarkably similar, and suggest a peak around  $\theta = 45^\circ$ . This agrees with the plots of  $E$  and  $\epsilon$  previously shown in Figures 5b and 5c, where the error peaked for flows with  $|\tan(\theta)| = 1$ . One possible explanation of this is that the ANN trained on steady state laminar flows is unable to account for the time dependence of  $\frac{D\mathbf{u}}{Dt}$ . Another more concrete explanation can be related to the model sensitivity analysis presented in Section IV.B.2. Because the model is likely over trained on the dependence of  $\mathbf{v}$ , many of the observed directional and magnitude errors can probably be attributed to this. Additionally, the errors  $E$  and  $\epsilon$  are shown to increase with  $St$ . This suggests that the non-dimensional data scaling is not similar enough for the ANN to make a meaningful prediction.

Additionally, to understand how the directional error is distributed, Figure 15 shows histograms of  $\beta$  by  $St$ . Figure 15a suggests that the majority of ANN-predicted accelerations,  $\mathbf{a}_{NN}$ , are orthogonal to the true particle acceleration  $\mathbf{a}$  for  $St = 0.05$ . For  $St = 0.5$  and  $5$ , however, the majority of  $\mathbf{a}_{NN}$  are closely aligned with the direction of  $\mathbf{a}$  as shown in Figures 15b and 15c. Each of these distributions also suggests a smaller peak around  $\beta = 90^\circ$ . This secondary peak could be observed because  $\frac{S}{f}$  and  $\frac{\Omega}{f}$  are positive while the ANN were trained with positive and negative values of  $e$  and  $\omega$ . Accordingly, multiple combinations of  $e$  and  $\omega$  collapse onto a single tuple of  $S$  and  $\Omega$  values in the louver flow

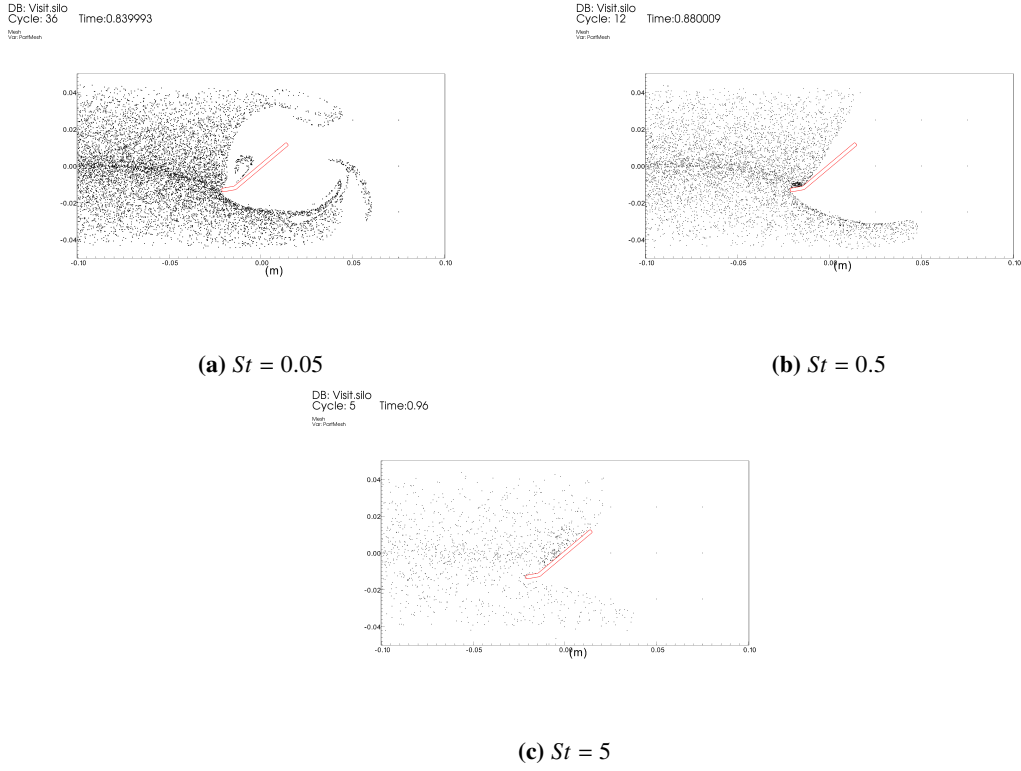


**Fig. 13 Particle interaction with flow around louver.**

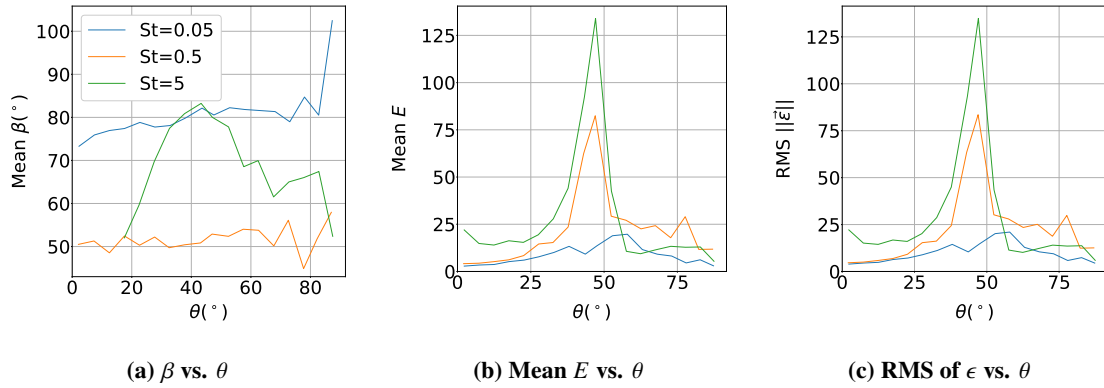


**Fig. 14 Master ANN error metrics when applied to particle data extracted from flow around the single louver. The blue, orange, and green lines correspond to  $St = 0.05$ ,  $0.5$ , and  $5$ , respectively.**

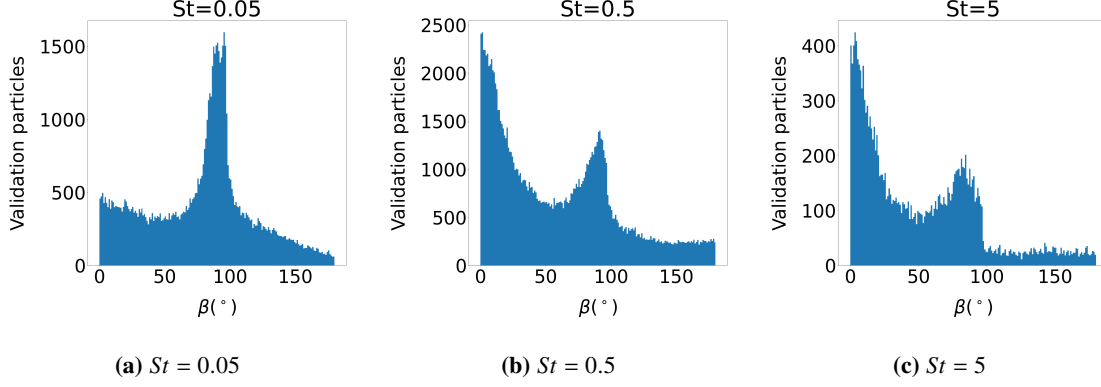


**Fig. 15** Distribution of  $\beta$  by  $St$  for particle data extracted from flow around the single louver. The bin widths are  $1^\circ$ .

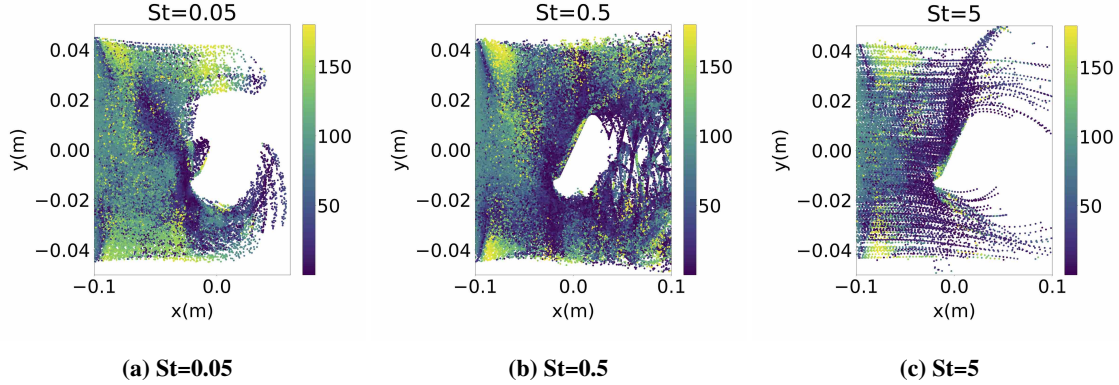


**Fig. 16** Scatterplots of  $\beta$  at the particle position  $\mathbf{x}_p$  by  $St$  for particle data extracted from flow around the single louver.

since the  $S - \Omega$  unit circle introduced in [2] is limited to the first quadrant ( $0^\circ < \theta < 90^\circ$ ).

Now that the average and distribution of directional error  $\beta$  has been characterized, it is of interest to understand the relationship between  $\beta$  and the particle position  $\mathbf{x}_p$ . Figure 16 shows how  $\beta$  depends on  $\mathbf{x}_p$  by  $St$ . The general trends indicate that the errors are higher near the inlet and near the top and bottom wall boundaries. As the particles approach the louver geometry and pass over or under it, the errors appear to decrease. It is not surprising that the ANN cannot predict the acceleration direction as well when near inlet, outlet, and wall boundaries because the canonical flows with which the ANN was trained are formulated as continuous flow fields free of any wall boundaries. It should also be noted that large errors are observed for those particles colliding with the louver or top/bottom walls. This is also not surprising because the ANN is not trained in any way to handle collisions. It is worth noting some implications of this formulation. For larger particles, the post-impact trajectory is driven primarily by the collision model. Since large particles behave ballistically, the velocity of the particle immediately after impact mostly predicts its trajectory. Hence ANN predictions show high accuracy in the post-impact region for the larger particles in Figure 16c, even though the ANN was not explicitly trained with collision data. Small particles respond more quickly to the flow field, and as such their post-impact trajectory will depend on both the collision model and the flow field. Hence lower accuracy was demonstrated for the trajectory prediction of these particles post-impact. Future work could explicitly incorporate the effect of collisions into the ANN formulation.

## VII. Conclusion

For laminar flow combining shear and vortex flows, the seven hidden-layer ANN presented in [2] was insufficient for making meaningful particle acceleration predictions. This led to the formulation of a custom loss function presented here that penalizes the ANN based on the  $L_1$  norm, or mean absolute value, of the normalized error vector  $\epsilon = \frac{\mathbf{a} - \mathbf{a}_{ANN}}{\|\mathbf{a}\|}$ . This loss function, along with manually limiting the learning rate  $lr$  allowed acceptable accuracy to be reached with the training and validation data. However, the model was not found to be generalizable as it could not serve as a suitable replacement for the acceleration governing equation. A model sensitivity analysis suggested that the ANN had been over-trained on the dependence of  $\mathbf{v}$ . The results from solving the governing equations with the ANN substituted for the true acceleration agree with this.

Finally, the particle-laden 2D CFD simulations of channel flow around the louver geometry only reaffirmed that the ANN is not generalizable in its current state. While the simulation results appeared to agree well with  $St$  theory, the ANN failed to accurately predict the acceleration using extracted particle data. However, for  $St = 0.5$  and  $5$ , the distribution of directional errors was most populated around  $0^\circ$ . While this is encouraging, the distributions exhibit a bi-modal distribution with a secondary peak around  $90^\circ$ , entirely orthogonal to the actual vector. The highest errors in the directional error prediction for any case appeared to be near the inlet or wall boundaries. This makes sense as the canonical flows with which the ANN was trained have no features similar to these.

While the ANNs presented are shown to fit the highly redundant and simplistic data set generated in this work, they are not generalizable. This could be implicitly because the training data fails to span the entire function space. The model sensitivity analysis supports the idea that the model was over-trained on particle features at  $t^* = 1.0$ , which is the time at which all of the training and validation data was taken. Alternatively, it could suggest that the method of flow decomposition presented here is fundamentally limited. In either case, further studies need to be conducted to elucidate why the model does not generalize.

## Acknowledgments

We would like to thank the Virginia Space Grant Consortium and the Sloan Scholars Mentoring Network for their funding and support of this research. The authors also acknowledge Advanced Research Computing at Virginia Tech for providing computational resources and technical support that have contributed to the results reported within this paper. URL: <https://arc.vt.edu/>

## References

- [1] Musgrove, G. O., Barringer, M. D., Thole, K. A., Grover, E., and Barker, J., "Computational design of a louver particle separator for gas turbine engines," *Proceedings of the ASME Turbo Expo*, Vol. 3, No. PART B, 2009, pp. 1313–1323. doi:[10.1115/GT2009-60199](https://doi.org/10.1115/GT2009-60199).
- [2] Bowman, T., Miranda, C., and Palmore, J. A., "Predicting Motion of Engine-Ingested Particles Using Deep Neural Networks," *ASME IMECE*, 2022.
- [3] NATO, "Recommended practices for the assessment of the effects of atmospheric water ingestion on the performance and operability of gas turbine engines," *AGARD Conference Proceedings*, 1995. URL <https://www.osti.gov/biblio/161497>.
- [4] NATO, "Erosion Corrosion and Foreign Object Damage Effects in Gas Turbines," *AGARD Conference Proceedings*, 1994.
- [5] Pan, H., and Render, P. M., "Impact characteristics of hailstones simulating ingestion by turbofan aeroengines," *Journal of Propulsion and Power*, Vol. 12, No. 3, 1996, pp. 457–462. doi:[10.2514/3.24057](https://doi.org/10.2514/3.24057).
- [6] Esmaily-Moghadam, M., and Mani, A., "Analysis of the clustering of inertial particles in turbulent flows," *Physical Review Fluids*, Vol. 1, No. 8, 2016, p. 084202. doi:[10.1103/PHYSREVFLUIDS.1.084202](https://doi.org/10.1103/PHYSREVFLUIDS.1.084202).
- [7] Maxey, M. R., "The gravitational settling of aerosol particles in homogeneous turbulence and random flow fields," *Journal of Fluid Mechanics*, Vol. 174, 1987, pp. 441–465. doi:[10.1017/S0022112087000193](https://doi.org/10.1017/S0022112087000193).
- [8] Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, Vol. 378, 2019, pp. 686–707. doi:[10.1016/J.JCP.2018.10.045](https://doi.org/10.1016/J.JCP.2018.10.045).

- [9] Seyed-Ahmadi, A., and Wachs, A., “Physics-inspired architecture for neural network modeling of forces and torques in particle-laden flows,” *Computers & Fluids*, Vol. 238, 2022, p. 105379. doi:[10.1016/J.COMPFLUID.2022.105379](https://doi.org/10.1016/J.COMPFLUID.2022.105379).
- [10] Minakowska, M., Richter, T., and Sager, S., “A Finite Element/Neural Network Framework for Modeling Suspensions of Non-spherical Particles: Concepts and Medical Applications,” *Vietnam Journal of Mathematics*, Vol. 49, No. 1, 2021, pp. 207–235. doi:[10.1007/S10013-021-00477-9/FIGURES/19](https://doi.org/10.1007/S10013-021-00477-9/FIGURES/19).
- [11] Cutnell, J. D., and Johnson, K. W., *Physics*, 9<sup>th</sup> ed., Wiley, 2012.
- [12] Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S., “Self-Normalizing Neural Networks,” *Advances in Neural Information Processing Systems*, Vol. 2017-December, 2017, pp. 972–981. doi:[10.48550/arxiv.1706.02515](https://doi.org/10.48550/arxiv.1706.02515), URL <https://arxiv.org/abs/1706.02515v5>.
- [13] Clift, R., Grace, J. R., and Weber, M. E., *Bubbles, Drops, and Particles*, Academic Press, Inc., New York, 1978.
- [14] Hindmarsh, A., and Petzold, L., “LSODA, Ordinary Differential Equation Solver for Stiff or Non-Stiff System,” , 2005. URL [https://inis.iaea.org/search/search.aspx?orig\\_q=RN:41086668](https://inis.iaea.org/search/search.aspx?orig_q=RN:41086668).
- [15] Matson, R. J., and Huggins, A. W., “The Direct Measurement of the Sizes, Shapes and Kinematics of Falling Hailstones,” *Journal of the Atmospheric Sciences*, Vol. 37, No. 5, 1980, pp. 1107–1125. doi:[https://doi.org/10.1175/1520-0469\(1980\)037%3C1107:TDMOTS%3E2.0.CO;2](https://doi.org/10.1175/1520-0469(1980)037%3C1107:TDMOTS%3E2.0.CO;2).
- [16] Capecelatro, J., and Desjardins, O., “An Euler-Lagrange strategy for simulating particle-laden flows,” *Journal of Computational Physics*, Vol. 238, 2013, pp. 1–31. doi:[10.1016/j.jcp.2012.12.015](https://doi.org/10.1016/j.jcp.2012.12.015).
- [17] Desjardins, O., Blanquart, G., Balarac, G., and Pitsch, H., “High order conservative finite difference scheme for variable density low Mach number turbulent flows,” *Journal of Computational Physics*, Vol. 227, No. 15, 2008, pp. 7125–7159. doi:[10.1016/J.JCP.2008.03.027](https://doi.org/10.1016/J.JCP.2008.03.027).
- [18] Miranda, C. J., and Palmore, J., “Predicting Erosion from Airborne Particles on Surfaces using a Soft-Sphere Collision Model,” *AIAA Aviation*, 2021. doi:[10.2514/6.2021-2636](https://doi.org/10.2514/6.2021-2636).
- [19] Brandt, L., and Coletti, F., “Particle-Laden Turbulence: Progress and Perspectives,” *Annual Review of Fluid Mechanics*, 2021. doi:[10.1146/annurev-fluid-030121](https://doi.org/10.1146/annurev-fluid-030121).
- [20] Kingma, D. P., and Ba, J. L., “Adam: A Method for Stochastic Optimization,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015. doi:[10.48550/arxiv.1412.6980](https://doi.org/10.48550/arxiv.1412.6980), URL <https://arxiv.org/abs/1412.6980v9>.
- [21] Gerhart, P. M., Gerhart, A. L., and Hochstein, J. I., *Munson, Young and Okiishi’s Fundamentals of Fluid Mechanics.*, 8<sup>th</sup> ed., Wiley, 2015. URL [https://books.google.com/books/about/Munson\\_Young\\_and\\_Okiishi\\_s\\_Fundamentals.html?id=ELp2CwAAQBAJ](https://books.google.com/books/about/Munson_Young_and_Okiishi_s_Fundamentals.html?id=ELp2CwAAQBAJ).
- [22] Desjardins, O., McCaslin, J. O., Owkes, M., and Brady, P., “Direct numerical and large-eddy simulation of primary atomization in complex geometries,” *Atomization and Sprays*, Vol. 23, No. 11, 2013, pp. 1001–1048. doi:[10.1615/AtomizSpr.2013007679](https://doi.org/10.1615/AtomizSpr.2013007679), URL <http://www.dl.begellhouse.com/journals/6a7c7e10642258cc,445ba8070f5787a9,341b457063bedf78.html>.
- [23] Borup, D. D., Elkins, C. J., and Eaton, J. K., “Experimental Analysis of a Particle Separator Design with Full-Field Three-Dimensional Measurements,” *Journal of Turbomachinery*, Vol. 142, No. 10, 2020, pp. 1–9. doi:[10.1115/1.4047112](https://doi.org/10.1115/1.4047112).