

Unsupervised Clustering and Performance Prediction of Vortex Wakes from Bio-inspired Propulsors

Alejandro G. Calvet, Mukul Dave, Jennifer A. Franck

Department of Engineering Physics, University of Wisconsin-Madison, Madison, WI, USA.

E-mail: jafraunck@wisc.edu

Abstract. An unsupervised machine learning strategy is developed to automatically cluster the vortex wakes of bio-inspired propulsors into groups of similar propulsive thrust and efficiency metrics. A pitching and heaving foil is simulated via computational fluid dynamics with 121 unique kinematics by varying the frequency, heaving amplitude, and pitching amplitude. A Reynolds averaged Navier-Stokes (RANS) model is employed to simulate the flow over the oscillating foils at $Re = 10^6$, computing the propulsive efficiency, thrust coefficient and the unsteady vorticity wake signature. Using a pairwise Pearson correlation it is found that the Strouhal number most strongly influences the thrust coefficient, whereas the relative angle of attack, defined by both the mid-stroke and maximum have the most significant impact on propulsive efficiency. Next, the various kinematics are automatically clustered into distinct groups exclusively using the vorticity footprint in the wake. A convolutional autoencoder is developed to reduce vortex wake images to their most significant features, and a k-means++ algorithm performs the clustering. The results are assessed by comparing clusters to a thrust versus propulsive efficiency map, which confirms that wakes of similar performance metrics are successfully clustered together. This automated clustering has the potential to identify complex vorticity patterns in the wake and modes of propulsion not easily discerned from traditional classification methods.

Keywords: bio-inspired propulsion, unsupervised machine learning, fluid dynamics

1. Introduction

Swimming and flying animals rely on the fluid around them to provide lift and/or thrust forces, leaving behind a distinct trail of vorticity. The structure and size of the vortex wake is a blueprint of the animal's kinematic trajectory, holding information about the forces and also the size, speed and direction of motion. This is true for a wide range of scales, from low to high Reynolds number swimmers and flyers, and includes man-made bio-inspired propulsive vessels or the vortex-induced vibration motion of bluff-bodied systems. Due to the diverse range of scales, kinematic motions, and varying objectives of organisms (e.g. speed vs. agility), the current classification of vortex wake modes available in literature is not inclusive. Rather than relying on predefined classifications of wake structure, this paper presents an unsupervised clustering process that automatically distinguishes between wake structures, and successfully groups them into regimes of similar propulsive performance.

The canonical bluff-body wake characterization is based on the model system of flow over an oscillating cylinder, whose wake properties are described and classified by Williamson and Roshko [1]. Using various amplitudes and frequencies of oscillation, their findings show regimes described by " $mS + nP$ ", where m is the number of single vortices (S) per cycles, and n is the number of clockwise/counter-clockwise pairs (P). The characterization is expanded to other systems, including that of oscillating foils, such as that done by Schnipper et al. [2] for the wake behind a pitching foil at Reynolds numbers of 220-440. The wake patterns were mapped based on pitching amplitude and non-dimensional frequency. Transition from a drag producing mode to a thrust generation mode was shown to be associated with a clear *reverse* von Kármán wake, where two alternating vortices are shed in a $2S$ mode with a counter-clockwise vortex at the top of the stroke and a clockwise vortex at the bottom.

The work by Schnipper et al. is preceded by important experimental findings such as a low-amplitude pitching foil experiment by Koochesfahani [3] which demonstrates that a combination of frequency and amplitude are responsible for specific vortex patterns and regimes of momentum excess or deficit. A similar observation is made by Lai and Platzer [4] for experiments on a plunging (heaving) foil in which the non-dimensional plunging velocity, a function of both amplitude and frequency, determined the threshold for thrust-generation. Connections between wake structure and kinematics are also shown in the motion of oscillating foils mimicking flapping flight of insects, in which the size and position of shed vortices are a function of frequency of oscillation [5]. Wake studies specific

to higher heave amplitudes in pitching and heaving foils have shown that a strong reverse von Kármán street is associated with higher thrust generation whereas a distorted wake is associated with degradation in thrust generation [6, 7].

However it has also been demonstrated that there is not always a one-to-one correspondence between thrust and wake structure. Floryan et al. give counterexamples that show large differences in the vortex wake structure with no change in performance, and conversely, that subtle changes can have great changes in performance [8]. Also mentioned are other factors that can decouple the wake from accurately predicting the performance including the Reynolds number, and the subsequent interactions of vortices within the wake which are unlikely to highly influence the drag and thrust on the foil.

More recently, wake structure classification has been augmented by machine learning utilities. Using information solely from a matrix of point measurements downstream of the wake, Colvert et al. [9] classify three specific wake patterns with an artificial neural network consisting of four layers. Vortex wake classification has also been performed by Wang and Hemati [10] where a k-nearest neighbors model classifies relative velocity time signals measured from a foil-based sensor. Some of the signals are known to correlate with well-known wake structures, however when the classification algorithm is applied to new exotic patterns, it produces high accuracy results. Li et al. [11] classified the wake patterns from computations of a fish-like flexible swimmer by training a fully connected neural network with one-time point measurements of velocity and vorticity in the wake.

The use of classification techniques such as the ones described above rely on the user to define the classes of wake patterns and select the appropriate class for specific kinematics. A classification model is subsequently trained with a subset of the data, known as the training set. The trained model is then utilized to sort kinematics from rest of the subset, known as the test set, which provides an estimate for the accuracy of the trained model. These methods would not be appropriate for discovery of chaotic or hybrid wake modes, for example those that result from multiple degrees-of-freedom or non-sinusoidal kinematics that may not be easily characterized with the $mS + nP$ notation, or may not be discernible by a human. As an example, the vortex wake patterns behind oscillating foils for energy harvesting applications are often generated with high-heave amplitudes and exotic wake patterns [12]. These challenges are exacerbated due to non-uniform or turbulent inflow conditions at high Reynolds numbers.

In lieu of classification, this paper uses *clustering*, which automatically groups similar samples without prior definition of the cluster labels and without supervised training. This method does not rely on the $mS + nP$ notion of counting vortices, nor does it have any pre-determined criteria defined in grouping similar wakes or kinematics together. The samples are images representing snapshots of wake vorticity, taken from computational fluid dynamics (CFD) simulations of pitching/heaving foils under various kinematic conditions. The clustering is performed with a k-means++ algorithm [13], and is supported by a convolutional autoencoder to reduce the the images to a latent space of pertinent features.

Figure 1 displays the two datasets of wake images that are analyzed. In order to tune and test the model, a prelabeled dataset is developed using the same oscillating foil kinematics as Colvert et al. [9]. This dataset represents wakes from computations of a one degree-of-freedom pitching foil and produces three distinct and well-defined wake patterns as shown in figure 1a. Subsequently, the tuned model is applied to vorticity wake images from a CFD database of thrust-generating oscillating foils with two degrees-of-freedom, pitch and heave, as shown in figure 1b. This larger and unlabeled dataset is comprised of more complex vortex wakes (see Dave et al. [14]) represented in figure 1b.

Using the autoencoder and clustering pipeline, the wake images are automatically sorted into groups. Since wake patterns are correlated with propulsive performance, the resulting cluster placement is expected to determine the performance regime in which the oscillating foil is operating based solely on the wake image. The clustering results are mapped onto a space of dominant kinematic variables and performance metrics (thrust and efficiency), demonstrating an automated sorting of the vortex images into distinct kinematic and propulsive regimes.

The paper is organized to describe the computational methods used to create the datasets, followed by a statistical correlation between the kinematics and performance metrics. The autoencoder and clustering methodology is then described, followed by the clustering results.

2. Computational Methods

2.1. CFD solver

Two databases are produced by simulating the flow field around an oscillating foil with CFD. The first is a *prelabeled dataset* formulated from a pitching foil with only one degree-of-freedom. The purpose of the prelabeled dataset is to tune hyper-parameters of the autoencoder and clustering on data with distinct wake structures easily classified with visual inspection. The main focus of the paper is on the second dataset, or the *unlabeled dataset*. This data is

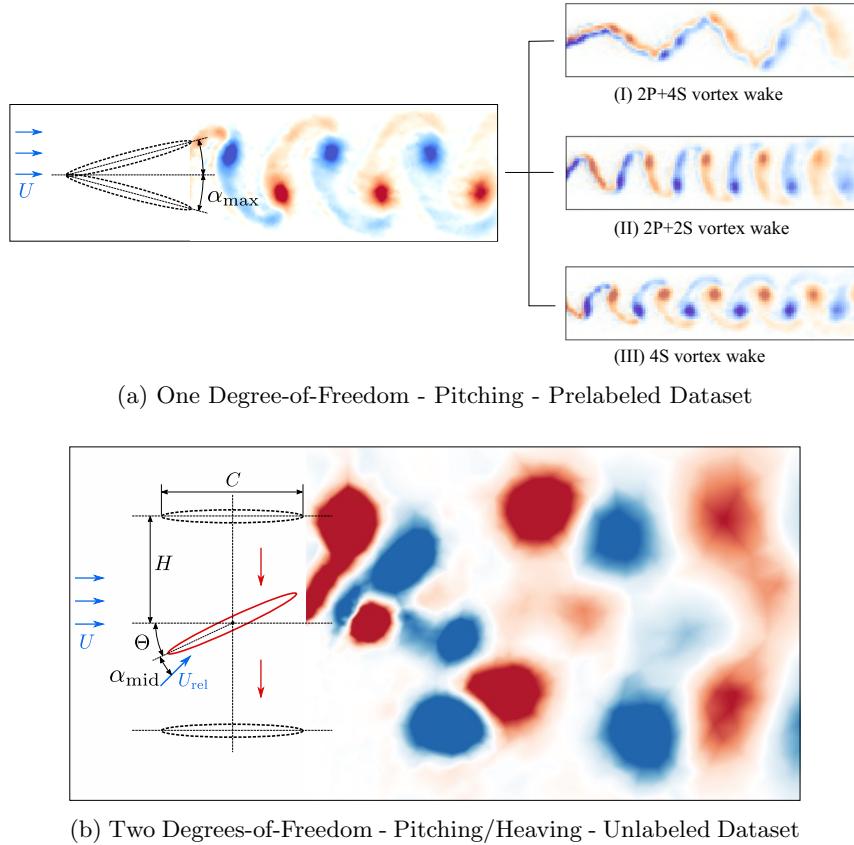


Figure 1: Oscillating foil propulsion datasets used for the clustering. **(a) One degree-of-freedom: pitching only.** This dataset generates three prelabeled vortex wake patterns that are a function of frequency (α_{max} remains constant). **(b) Two degrees-of-freedom: pitching and heaving.** This dataset generates a range of complex wake patterns by varying pitch amplitude, heave amplitude, and frequency.

comprised of flow fields around a pitching and heaving foil with two degrees-of-freedom, generating more complex and chaotic wake structures over a much wider range of kinematic flapping parameters. Once the CAE and clustering algorithm have been developed and tuned using the labeled dataset, the analysis methods are subsequently applied to the larger unlabeled dataset.

Simulations are performed with a second-order accurate open-source finite volume solver from the *OpenFOAM* package. To solve for the fluid flow around a moving foil, a dynamic mesh library was utilized that dynamically computes the displacement of each mesh element, \mathbf{x}_m , at every time step driven by the motion of the oscillating foil. Kinematics of the foil are translated to a rigid body motion of each element on the surface of the foil, and implemented as a boundary condition. The mesh motion equation is derived from the linear elasticity equation of motion [15], and includes a diffusivity constant that scales inversely with distance from the foil, such that the cells in the immediate vicinity of the body are not dramatically skewed [16]. Figure 2 displays the two-dimensional computational domain, which is 100 chord lengths (100C) in each coordinate direction, with inlet and outlet boundary conditions as denoted in the figure.

2.2. Prelabeled dataset: Pitching foil

This dataset is obtained by solving the incompressible Navier-Stokes equations of flow over a NACA0012 foil as it pitches about the leading edge. The simulation is performed at a Reynolds number, $Re = UC/\nu = 1000$, where ν is kinematic viscosity of the fluid, U is the freestream velocity, and C is chord length of the foil. The specific kinematics for this prelabeled data is chosen to replicate those simulated and previously classified by Colvert et al. [9]. The 15 simulations, each at a different kinematics, can be classified by three distinctly identifiable wake patterns using the “ $mS + nP$ ” notation.

As shown in figure 1a, the maximum pitching amplitude, α_{max} , is held constant at 5.7° . The kinematics are varied by changing the frequency of oscillation, F , which is non-dimensionalized as the reduced frequency,

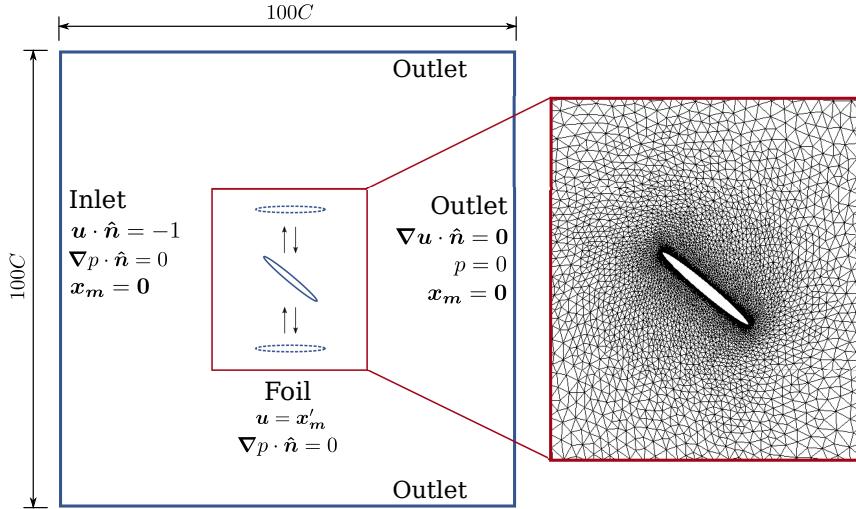


Figure 2: Simulation domain (not to scale) and mesh configuration during the upstroke.

Table 1: The prelabeled dataset contains 15 simulations, 5 from each regime, that were performed by varying frequency of a foil pitching about its leading edge. These cases generate three distinct vortex wake patterns shown in figure 1a.

Prelabeled dataset	
Wake Pattern	F^+ range
2P + 4S	0.317 – 0.383
2P + 2S	0.525 – 0.558
2S	0.592 – 0.658
Simulation Properties: $Re = 1000$; $B/C = 0$; $\alpha_{max} = 5.7^\circ$	

$F^+ = FC/U$. Fifteen frequencies are tested, resulting in fifteen unique kinematics that can be sorted into three classes of wake patterns visualized in figure 1a and listed in table 1. The low frequency cases exhibit a $2P + 4S$ mode, the intermediate frequency range has a $2P + 2S$ mode, and the highest frequencies tested exhibit a $2S$ mode, or a reverse von Kármán vortex street.

The mesh for the low Reynolds number simulations in this dataset is comprised of 74,728 cells, with 1022 cells defining the foil surface. The mesh size within the boundary layer is 0.002 chord lengths, and is gradually increased radially away from the foil. The mesh resolution of the wake is adequate to clearly resolve three wavelengths of the vortex wake at the lowest flapping frequency.

2.3. Unlabeled dataset: Pitching and heaving foil

The second, unlabeled dataset is designed to produce a richer variety of propulsive wake structures. To accomplish this, sinusoidal motion of a 10% thick ellipse is prescribed in both the pitch and heave directions, with variations in pitch amplitude (Θ), heave amplitude (H), and frequency as depicted in figure 1b and outlined in table 2. The phase difference between the heave and pitch stroke, ϕ , is 90 degrees such that the foil is at zero pitch at the top and bottom of its heave stroke and at maximum pitch at mid-stroke. Due to the added degree of freedom in the heave direction, the Strouhal number, $St = 2HF/U$, is used as a non-dimensional parameter representing the maximum heave velocity. A function of both the frequency and heave amplitude, St has been shown to be correlated with propulsive performance in literature [3, 4, 17]. High-heave kinematics ($H/C \geq 1$) are the predominant focus of the unlabeled data, as they are found to add a high degree of variability into the wake structure and are far less studied in previous literature. However, an additional set of kinematics at lower heave amplitude ($H/C = 0.75$) and different pitch axis location, $B/C = 0.33$, is also included in the statistical correlation analysis in section 3.

Due to the changing relative velocity vector, the relative angle of attack of the flow with respect to the foil, α , varies over the cycle as demonstrated by figure 3. As noted in previous work [14, 6, 7], the relative angle of attack can be used as an indicator of propulsive performance. Over the span of the high heave and high pitch kinematics tested, the shape of α throughout the cycle can take various forms, and thus two critical values are reported for

Table 2: The unlabeled dataset contains 63 different combinations of pitch amplitude (Θ) and frequency (St) for each of two heave amplitudes (H) to produce 126 unique sets of kinematics.

Unlabeled dataset		
H/C	Θ range	St range
1.0	$15^\circ - 55^\circ$	$0.2 - 0.6$
2.0	$30^\circ - 70^\circ$	$0.4 - 1.2$
Simulation Properties: $Re = 10^6$; $B/C = 0.5$; $\phi = 90^\circ$		

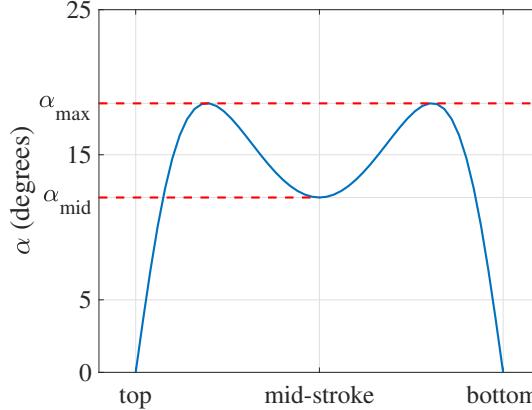


Figure 3: Variation in the relative angle of attack during a downward stroke of the foil for $H/C = 1$, $St = 0.6$, $\Theta = 50^\circ$.

each set of kinematics: the maximum value, α_{max} , and value at mid-downstroke, α_{mid} .

An unstructured mesh is generated with approximately 80,000 cells, with a high density clustered close to the foil to ensure adequate boundary layer resolution. To demonstrate the mesh structure and resolution in close vicinity of the foil, figure 2 illustrates the foil position and deformed mesh configuration during upstroke. The maximum value of non-dimensional wall distance, y^+ , at the first mesh layer on the foil is 1.87 when the foil is stationary at zero angle of attack, and the number of circumferential mesh cells along the foil boundary is 6774. A mesh resolution study has been previously performed [14] where it is demonstrated that the unsteady forces and thrust generation are adequately resolved at the current resolution. Furthermore, the wake vorticity is compared between the current mesh and one with higher resolution in the far-field. The comparison highlights that the same vortex structure persists, although the strength of the vortices is weakened with the slightly lower mesh resolution. This has a similar effect as varying the RANS turbulence model which is discussed in more detail below. The time step for the computation is determined dynamically by imposing a maximum Courant number of 0.95.

This dataset is performed at a high, turbulent Reynolds number of 10^6 using a Reynolds-averaged Navier-Stokes (RANS) solver with $k-\omega$ SST turbulence closure. The $k-\epsilon$, realizable $k-\epsilon$, and Spalart Allmaras models were also tested for a single set of kinematics to examine model sensitivity. The $k-\epsilon$ model was too dissipative and the thrust computed, $C_T = 1.22$, deviated significantly from the other three models, which were very close to each other with $C_T \approx 1.42$. Figure 4 compares instantaneous vorticity fields from computations using the four models. Except for $k-\epsilon$, there are only minor differences among the remaining three models in terms of the vortex formation during oscillation, and the overall structure and location of vortices aligns well. It is expected that these small differences would be further diminished once the images are processed by the autoencoder and reduced to a latent space for clustering. More details on this dataset including the simulation set-up, validation, mesh refinement, and comparison against experimental data, are outlined in Dave et al. [14].

2.4. Performance metrics

The kinematics chosen for the unlabeled dataset fall within the thrust-producing regime, in which the net propulsive force created by the foil's motion can be characterized by the thrust coefficient,

$$C_T = \frac{\langle F_x \rangle}{\frac{1}{2} \rho U^2 C}, \quad (1)$$

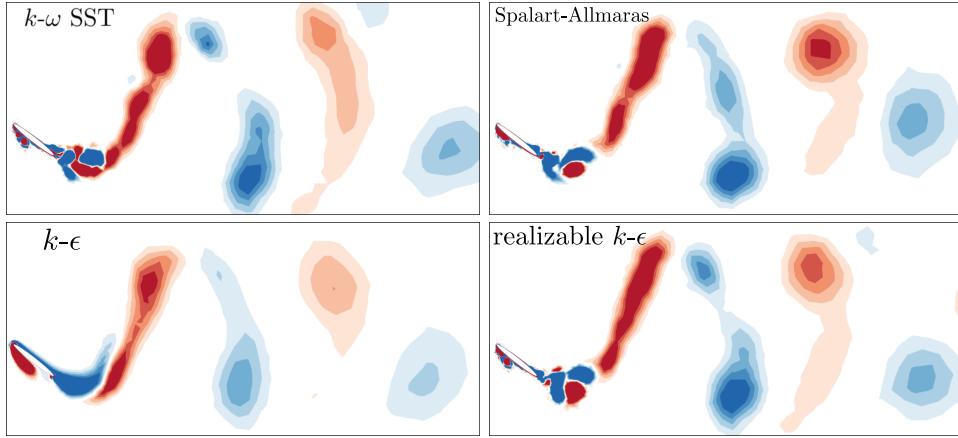


Figure 4: Instantaneous spanwise vorticity ($\Omega C/U$) fields from computations with four different turbulence models for the kinematics $H/C = 1$, $St = 0.6$, $\Theta = 45^\circ$. Contours range from -2 (blue) to 2 (red).

where $\langle F_x \rangle$ is the time-averaged force in the horizontal direction, opposing the freestream flow. There is power exerted to generate thrust, characterized by the input power coefficient, C_P . The power coefficient is computed as the work required to perform the vertical and pitching motions, or

$$C_P = -\frac{\langle F_y(t) \frac{dh(t)}{dt} + M_z(t) \frac{d\theta(t)}{dt} \rangle}{\frac{1}{2} \rho U^3 C}, \quad (2)$$

where $F_y(t)$ and $M_z(t)$ are the vertical force and the moment during the heave, $h(t)$, and the pitch, $\theta(t)$, motions. The time-averaged input power will be denoted by C_P . Furthermore, the amount of input power that is converted to thrust is defined by a propulsive efficiency,

$$\eta = \frac{C_T}{C_P}. \quad (3)$$

3. Correlation of Kinematic Parameters with Performance Metrics

In order to better correlate parameters that most strongly influence the variation in performance and hence the wake structure, statistical analysis is performed on the unlabeled two degrees-of-freedom dataset. Three independent variables, Θ , H/C , and St , are varied directly, and the secondary kinematic parameters, α_{mid} and α_{max} , are computed. For performance metrics, the propulsive efficiency, η , the power input, C_P , and thrust coefficient, C_T , are all evaluated.

To discover which kinematic parameters most strongly influence performance, a matrix of Pearson correlation coefficients is computed in figure 5a. This matrix displays a pairwise statistical correlation between two individual variables in terms of magnitude and direction of the association. Values close to 1 are directly correlated, values close to -1 are inversely correlated, whereas values close to 0 do not show a correlation at all.

As a result of this matrix, the inverse correlation between C_T and η is observed here as is also well established in literature. Additionally, it is seen that the most influential parameter to increase C_T is an increase in St , which is consistent with other findings e.g. [17, 18, 19]. The input power, C_P , is also strongly dependent on St , but not to the same degree as the thrust coefficient. The increase of thrust and input power with St has been explained using scaling relations by Floryan et al. [20]. This work shows an increase in η with respect to amplitude at high St , but the opposite trend with low St , explained by the influence of drag. These scaling laws are relevant only for lower relative angles of attack, or kinematics that do not display large flow separation.

In contrast, the current research explores a frequency range and heave amplitude range that commonly result in high relative angles of attack inducing massive flow separation. For the kinematics explored the pitch amplitude has a slightly positive correlation with η however there is no correlation between heave amplitude and η . Hence it is observed that the thrust coefficient increases with the heave amplitude at no cost to the propulsive efficiency in an average sense. To independently establish the role of heave compared with pitch, a more systematic approach would have to be employed that holds St constant while increasing heave over a wider range of amplitudes. The relative angle of attack is also important in prediction of the output kinematics, as shown by many others in the literature e.g. [18]. Figure 5a confirms these findings by demonstrating that a decrease in either α_{max} or α_{mid} has

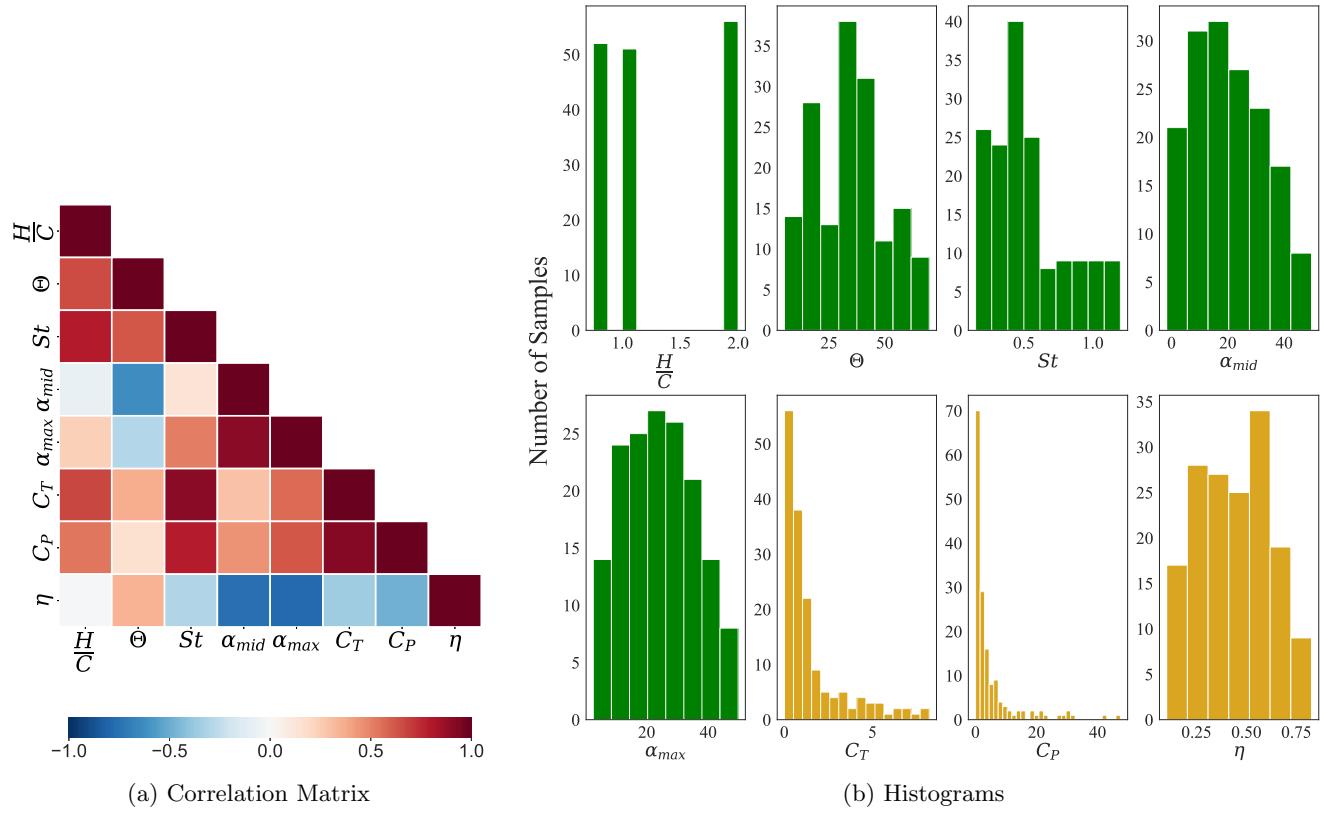


Figure 5: (a) Correlation Matrix: Pearson correlation coefficients among the kinematic parameters and output metrics, with a value of -1 indicating inverse correlation and 1 indicating direct correlation. (b) Histograms: Frequency of samples in different bins or range of values for each parameter studied. In green - kinematic parameters, in gold - performance parameters.

the strongest impact on increasing η . In terms of deciphering the effects of α_{max} versus α_{mid} , the correlation matrix demonstrates the strong inverse relationship of α_{mid} with Θ , whereas α_{max} has a strong proportional relationship to St for the kinematics range explored.

It is important to note that the statistical correlations of figure 5a are valid only for the range of parameters available in the unlabeled dataset. Thus, figure 5b shows the data sample distribution for each of the kinematic parameters in green and the output metrics in gold. The kinematic parameters are fairly well-distributed with the exception of heave amplitude, H/C , in which only three distinct values are explored. The output metrics C_T and C_P are more heavily concentrated at low values, whereas propulsive efficiency has a wider distribution. Including more data at lower heave amplitudes may skew the parameters and correlations in a slightly new direction, however this paper specifically focuses on the thrust-producing kinematics at high heave amplitudes. The narrow focus on high heave kinematics imposes a tractable limit to the number of simulations while still generating a variety of wake modes for the clustering.

4. Autoencoder and Clustering Methodology

4.1. Data pre-processing

The input into the convolutional autoencoder are images of the spanwise vorticity fields of the wake behind the oscillating foil. Vorticity is plotted in the *Tecplot* software, with 5 contours from blue (RGB 0, 0, 255) to red (RGB 255, 0, 0). The window size, kept constant throughout the dataset at $8.5C$ by $11C$, is large enough to accommodate the highest heave cases with the foil stroke centered vertically, and aligns the trailing edge of the foil on the left border of the frame. The horizontal frame width is large enough to capture at least one wavelength of the resulting wake for the lowest oscillation frequency. A final post-processing script floods the zero-vorticity background to black (RGB 0,0,0) and reduces the resolution to 128 by 128 pixels for the final version of the PNG files used by the autoencoder.

The prelabeled dataset contains 15 unique kinematics, each sampled with 14 frames, for a total of 210 images. The unlabeled dataset samples each unique set of kinematics 21 times over a half stroke. The heave amplitudes of $H/C = 1$ and $H/C = 2$ are considered in the clustering algorithm with 121 unique kinematics, resulting in 2541 image samples. It is important to note that each of these image samples is clustered independently and not as a group with the other 20 samples from the same kinematics.

4.2. Autoencoder architecture

The clustering methodology consists of a deep convolutional autoencoder (CAE) [21] used for image feature extraction followed by a k-means++ algorithm to cluster similar images in the embedded feature space. The convolutional autoencoder described below is implemented using the Python based libraries *TensorFlow* [22] and *Keras* [23].

An autoencoder falls within the broader category of neural networks, however it often has the specific goal of reducing a complex image to a compact representation of the most dominant features [24, 25], which in this case is the fundamental vortex wake structure. Pertinent features within the image are detected with filters, which create feature maps through a convolution operation on the preceding layer. Each convolutional layer utilizes multiple filters, hence generating multiple feature maps with the same height and width, and the number of maps defines the depth. Additionally, the max-pooling layers perform a downsampling operation with a 2×2 filter along the width and height dimensions, resulting in a reduction of factor 2 in the size of each dimension, while the depth remains constant.

The combination of convolutional and max-pooling layers increases the neural network performance through many strategies: reducing the feature space to a more manageable size, reducing the number of parameters in the network to control for overfitting, making the CAE invariant to small distortions and translations in the input image, and helping the CAE to arrive at an almost scale invariant representation of the input images (detecting features irrespective of their size in the image).

Figure 6 displays a summary of the architecture implemented. The first portion is the convolutional encoder, which consists of five sequences of convolutional and max-pooling layers that compress the original 128×128 pixel image with 3 color channels to a 4×4 feature map with 4 channels. The next step is to flatten the 3-dimensional matrix, followed by a fully-connected neural network layer to yield a 10 dimensional feature space, commonly known as the bottleneck. The convolutional decoder, which consists of layers in reverse order with exactly the same hyperparameters as used in the encoder, reconstructs the image. Finally, outputs are compared pixel to pixel with the original image inputs using a cross-entropy loss function [26, 27].

The *Adam* optimization algorithm [28] is utilized to iteratively compute the parameter values for the convolution filters as well as the fully connected layers, so as to decrease the loss function and mimic the input matrix as accurately as possible. The number of layers, the filter size, the number of filters in each layer, normalization functions, and use of regularization methods have also been tuned to reduce the loss function while increasing the bottleneck compression. Since the input image is compared with the reconstructed image to reduce the loss function and extract the dominant features at the bottleneck, this step is considered unsupervised. Appendix A shows a detailed scheme of the final CAE architecture.

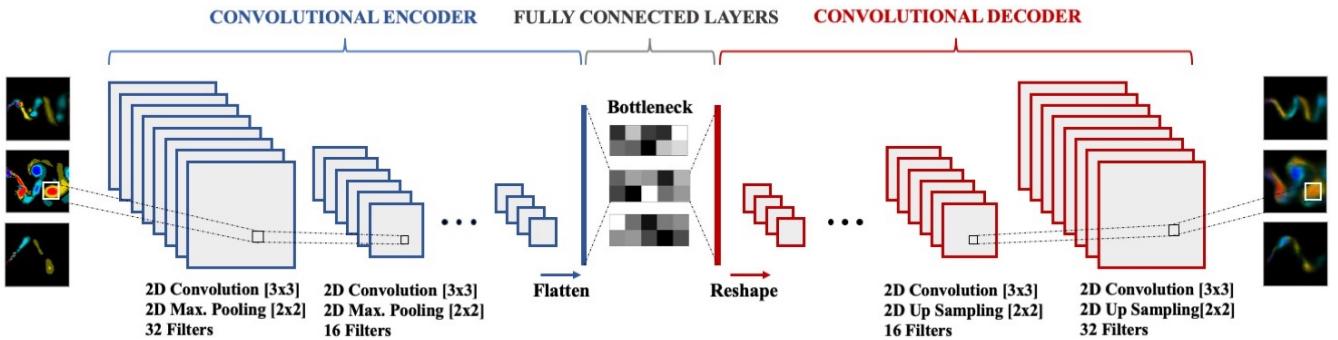


Figure 6: Convolutional Autoencoder architecture.

Two indicators of a good autoencoder are a high compression from the input image to the bottleneck and an accurate reconstruction of the original image. A high compression means a low dimensional feature space as an input to the clustering algorithm, whereas a good reconstruction ensures that those features represent enough of the original image. The balance between high compression and proper reconstruction is not easy to achieve,

especially when the original images have high resolution data resulting in a deeper neural network in order to achieve a high compression. In the current CAE, a strategy based on skip-connections proposed by Xiao-Jiao Mao et al. [29] was implemented to address this challenge. Skip-layer connections link convolutional and their respective deconvolutional layers, increasing both training speed and reconstruction accuracy. This method allows the loss function signal to back-propagate to bottom layers, directly reducing the problem of vanishing gradients [30] while keeping a high compression level. It also enables the encoding layers to pass information to the decoding layers, hence increasing the image reconstruction accuracy.

4.3. *k-means ++ clustering*

Once the data is properly compressed into a low dimensional latent space of features, the resulting arrays are automatically grouped using a *k-means++* clustering algorithm. This method was chosen after comparing with results from more sophisticated methods including K-medoids [31], hierarchical clustering [32], and DBSCAN [33], none of which showed any improvement over *k-means++*. The clustering algorithm is implemented using the *scikit-learn* library [34].

k-means [35] is an iterative method that requires user input to determine N , the number of cluster groups. The algorithm subsequently initializes N random centers, which are points in the multidimensional feature space. For each data point (in the compressed latent space), the distance to each center is calculated and the point is classified into a group based on minimum distance. Based on the classified points, the cluster center is recalculated with each iteration by taking the mean of each dimension of all the vectors in the group. *k-means++* is an enhanced version of the original *k-means* algorithm which improves the initial choice of centers, increasing both accuracy as well as speed [13].

Thus, the number of clusters, N , is predetermined by the user. The preferred number of clusters will be the N value that has a minimum mean intra-class distance and a maximum mean inter-class distance, which are not always concurrent with one another. The elbow method [36] computes the total within-cluster sum of square (WSS) error, known as distortion, and plots it against N , choosing the value of N that is on the ‘elbow’ of the curve as the indicator of the approximate number of clusters. The average silhouette score method [37] is often used in tandem with the elbow method to measure the quality of a clustering. This method determines how well each point lies within its cluster by estimating *cohesion* and *separation* as Euclidean or Manhattan distances. For an individual point, cohesion is the average distance of that point to all the other points in its cluster, and separation is the minimum of its average distance to all the points in another cluster. The Silhouette score is a combination of both factors, typically ranging from 0 to 1, where a higher score indicates better clustering.

5. Results and discussion

5.1. Clustering accuracy of prelabeled dataset

The one degree-of-freedom dataset described in figure 1a is comprised of a foil in a pure pitching mode with varying non-dimensional frequency, F^+ . The resulting database has 15 different kinematics producing wakes that have been prelabeled into three distinct groups based on the vortex structure [9], with 14 images of the wake sampled for each frequency providing a total of 210 images. The first row in figure 7 demonstrates vortex patterns typical of each clustering group, and the input image into the CAE.

The clustering is performed on the encoded latent space of features produced by the CAE algorithm, and demonstrated in the second row of figure 7. The encoded arrays shown are each significantly different from one another, each representing a different wake pattern and thus providing the most important features for the *k-means++* clustering algorithm. The third row of figure 7 depicts the reconstructed image and demonstrates that the tuned CAE is able to achieve both a high compression level in the encoding and an accurate image reconstruction.

The CAE was trained with 138 samples from this dataset, followed by 42 samples for tuning the hyperparameters and 30 samples for validation. The 210 images were then compressed and the resultant latent space of features was used as input for the *k-means++* clustering. In this prelabeled dataset, the number of clusters is known and was set to $N = 3$.

The clustering algorithm grouped exactly 70 samples in each of the 3 clusters. All the predicted groupings were coincident with the visually inspected data implying a 100% accuracy for this small dataset, summarized by the confusion matrix in figure 8.

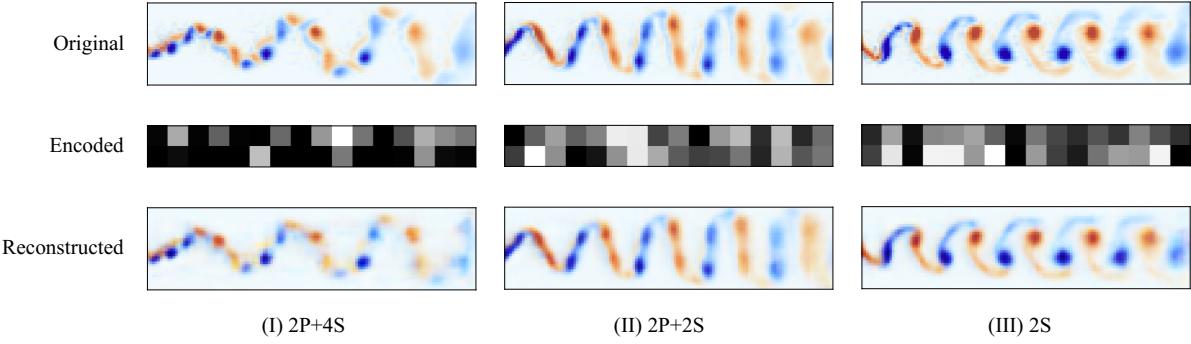


Figure 7: Three different vortex wake patterns for the one degree-of-freedom samples, presorted into categories I, II, and III. Representation of the original images (first row), encoded images or latent space of features (second row), and reconstructed images (third row).

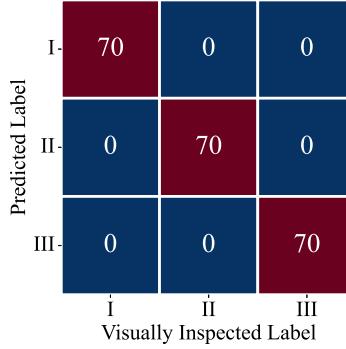


Figure 8: Confusion matrix for the prelabeled one degree-of-freedom dataset.

5.2. Clustering details of unlabeled dataset

The two degrees-of-freedom model introduced in figure 1b produces a more complex wake structure due to the combined heaving and pitching motion and the higher Reynolds number, resulting in 121 unique kinematics and a total of 2541 samples of vorticity wake images, that are clustered with the k-means++ algorithm. Two different strategies are implemented to analyze the clustering performance as described in the sections below.

5.2.1. Validation using visually inspected clustering with $N = 4$ All of the 121 kinematics were visually labeled and placed in four different groups based on similar vortex wake patterns. A subset of 504 image samples from the total of 2541 samples are subsequently labeled. The samples are carefully selected such that there are samples from each of the unique kinematics, and that there are equal number of samples from each group.

The CAE is trained with the remaining 2037 samples and subsequently used to cluster the 504 visually labeled samples. The cluster prediction is compared against the visually inspected labels through the confusion matrix shown in figure 9a.

The confusion matrix demonstrates that all the samples visually labeled as cluster A coincide with the clustering predictions. However, the results for the other 3 predicted clusters differ from the visually determined labels. The samples in cluster B are sorted into clusters A, B and C, and the samples from clusters C and D are also intertwined with each other. Some examples of the clustering results are shown in figure 9b which consists of 5 random samples from each of the four predicted clusters. The samples C4 and C5, for instance, do not exhibit a harmonic vortex wake pattern similar to those in C1, C2, and C3. However, due to the increased vorticity, the algorithm finds more similarities with the samples in cluster C than the samples in cluster A or B which do not have a clear wake structure.

Even though figure 9a shows that the predicted labels do not match the visually assigned labels, figure 9b demonstrates that most of the samples grouped together are quite similar. The discrepancies between predicted and visually determined labels seem to be a consequence of the criteria for sorting (i.e. visual sorting focused on wake structures whereas the CAE and the clustering algorithm placed a stronger value on the wake height and

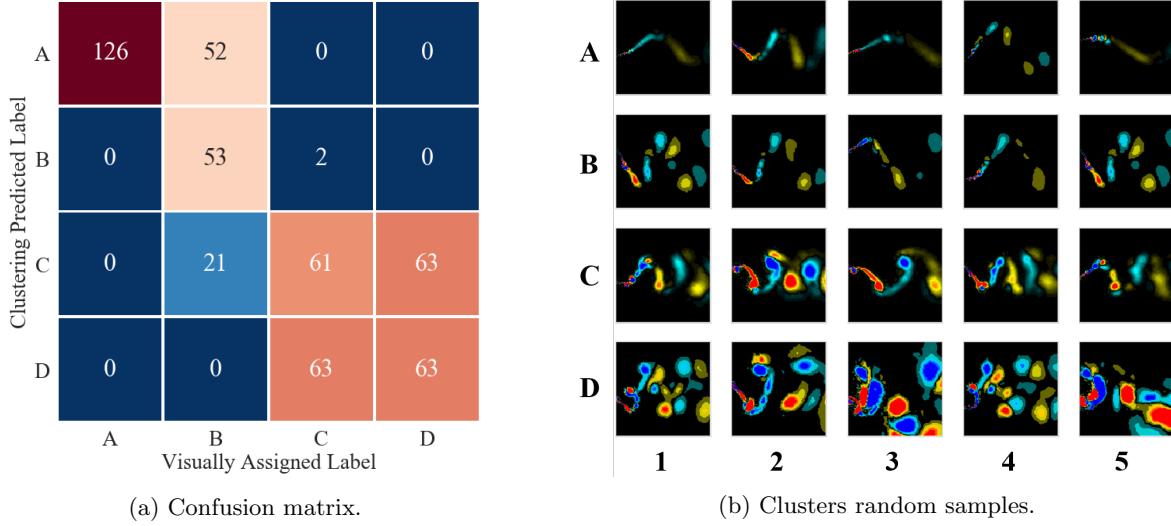


Figure 9: Results for clustering with $N = 4$. (a) Confusion matrix. Predicted clustering compared with visually assigned labels. (b) Random samples from each predicted cluster.

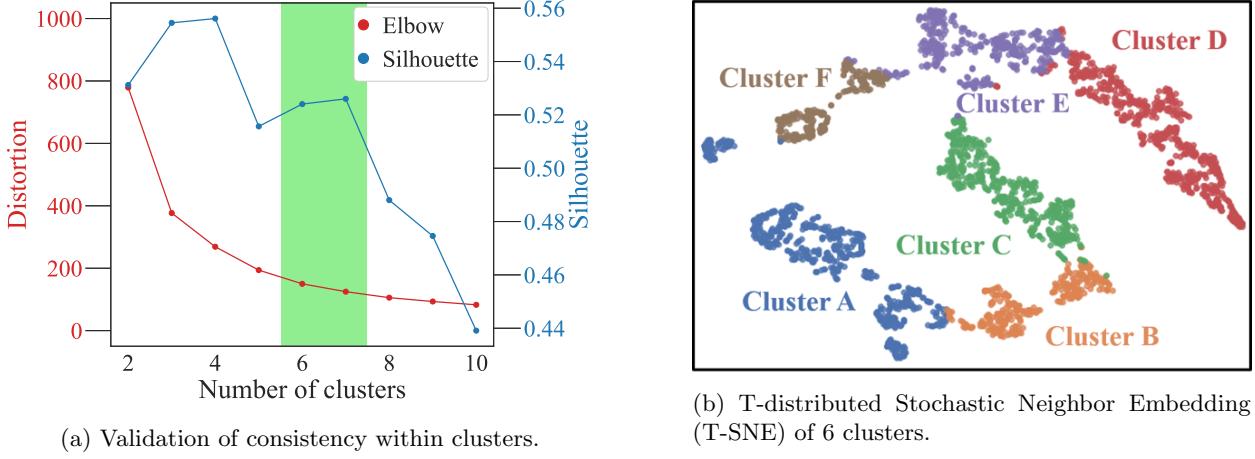


Figure 10: The optimal number of clusters is determined by a combination of the silhouette and elbow methods, shown in (a). The optimal region is shaded green indicating $N = 6 - 7$ being the optimal number of clusters for this dataset. The T-distributed stochastic neighbor embedding (T-SNE) distribution is shown in (b) for the 6 clusters.

amount of vorticity) rather than a bad algorithm performance. It also indicates that a different number of wake pattern groups may yield a clearer clustering.

5.2.2. Clustering with $N = 6$ As a result of the discrepancies between visually assigned and predicted labels with $N = 4$, the number of clusters was increased to account for some of the differences unable to be distinguished by only four bins. Two techniques introduced in section 4.3, the elbow and silhouette methods, were used together to determine the optimum number of clusters for the dataset. Figure 10a plots the distortion and silhouette values versus the number of clusters, and the light green region highlights the area that both minimizes distortion and finds a local peak in the silhouette value, indicating that $N = 6 - 7$ is the optimal number of clusters. In order to visually display the distribution of the latent space, a T-distributed stochastic neighbor embedding (T-SNE) is applied to the resultant latent spaces in order to further reduce the dimensions into a bi-dimensional embedding. Figure 10b displays the bi-dimensional embedding of the 2541 latent spaces of features encoded by the convolutional autoencoder, showing the samples which were grouped in each of the 6 different clusters. The number of clusters was set to 6 for simplicity since $N = 7$ did not significantly improve the final results.

Figure 11 displays 10 random samples from each predicted cluster. The vortex wakes that are grouped together in the same cluster all exhibit a similar pattern. Clusters A and B group samples with low vortex wake intensity.

Cluster B differs from cluster A in that the heave, H/C , is greater resulting in a stronger vorticity signal in the samples.

Clusters C and D both group samples with more dramatic flow separation than the previous clusters, and both with $H/C = 1$. The high oscillating frequency also decreases the wavelength between vortices, making these samples easy to distinguish. For cluster D, vortices in the wake are stronger and the wake pattern is more coherent relative to cluster C. A clear reverse von Kármán street is also observed likely due to a smoother variation of α , in contrast with cluster C that has a more chaotic wake due to a distorted profile of α as shown in figure 3. One exception is sample D9 which is similar to those in cluster C, and may represent a transition between the two clusters. Finally, clusters E and F show chaotic vortex wake structures at the larger heave amplitude, where there is not a clear pattern defined. Cluster F contains samples with higher vorticity and higher oscillation frequency (as observed from wavelength of the wake).

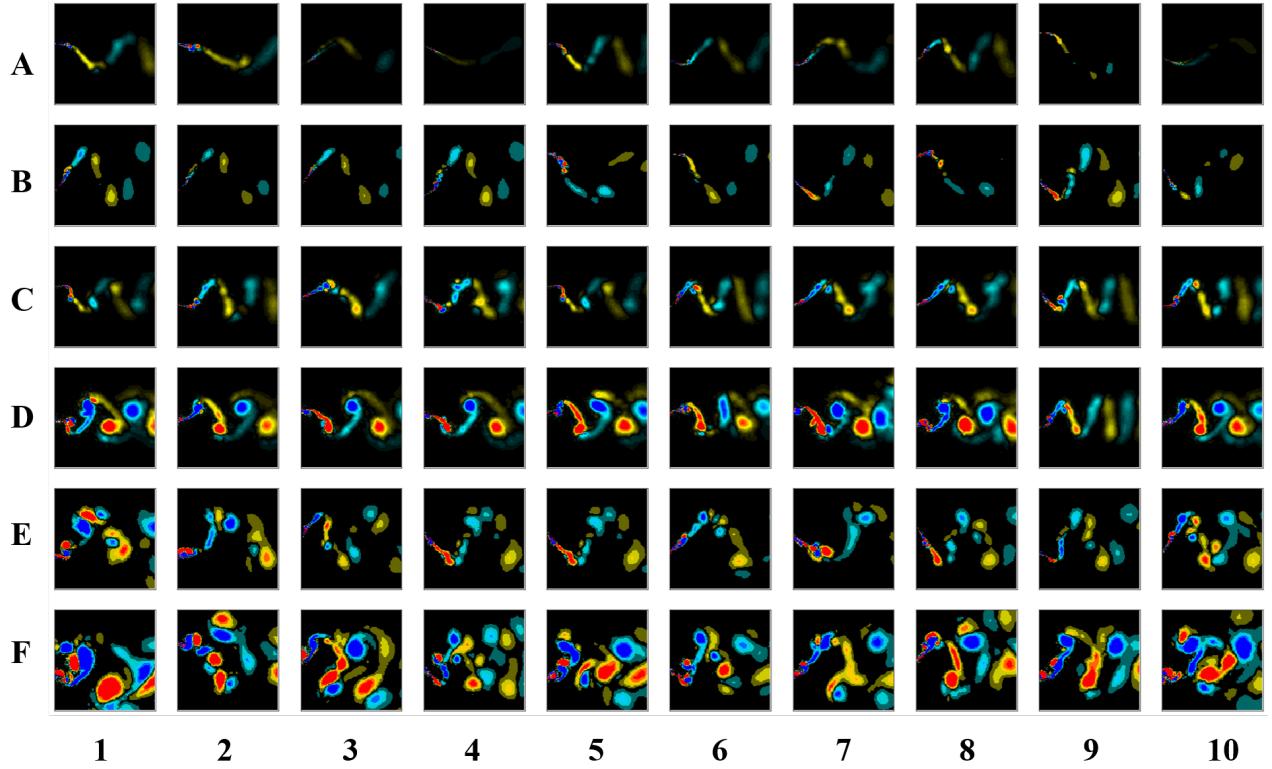


Figure 11: Each row shows 10 random samples from a particular cluster. Most samples clustered together show a similar pattern, though there are exceptions such as sample 9 in Cluster D, which is more similar to the patterns shown in cluster C.

5.3. Correlation of clustering with kinematics and propulsive performance

A method to evaluate the effectiveness of the clustering is to investigate its correlation with the propulsive performance, as it has been observed that propulsive thrust generation is strongly correlated with vortex wake structure [6, 7]. In figure 12, each cluster is mapped onto the kinematics space in terms of St and α_{mid} , and the performance space in terms of C_T and η . In the kinematics space, each cluster corresponds to a rather unique region with little overlap.

Essentially the clusters A, C, and D group the kinematics with $H/C = 1$ in order of increasing frequency (also demonstrated by the increase in α_{mid}) whereas clusters B, E, and F do the same for kinematics with $H/C = 2$. The wake height is hence the most prominent feature used for clustering the samples, which are then sorted into three groups based on the spread of vorticity and the pattern in the wake. Overlaps in regions of performance space are most prevalent between clusters B and C, and another region between D and E. However if both heave amplitudes are observed separately, there is minimal overlap among the clusters A-C-D and among B-E-F.

Clusters A and B are a region with low α_{mid} and relatively low St in the kinematics space. The separation

between the two clusters in both the kinematics and performance space is strong, with the primary separation likely due to the two distinct heave amplitudes. The wake patterns are characteristic of the low α_{mid} values indicating that it remains below the static stall angle of attack and there is minimal to no boundary layer separation.

Both clusters A and B show low C_T and high η in the performance space, which is in agreement with the correlations shown in figure 5a where it was discussed that α_{mid} and St are directly correlated with C_T but inversely correlated with η . Cluster C has a higher α_{mid} than both A or B, and a St range in between the A and B. This causes less than ideal performance in which the efficiency drops significantly and the thrust coefficient is between clusters A and B. Cluster D is similarly characterized, although it is uniquely placed in the kinematics space, its performance has low efficiency and low non-dimensional thrust, and shares some overlap with cluster E in terms of performance. Finally, clusters E and F, which have the highest St are correlated with the highest thrust generation. Cluster E offers an intermediate range of both efficiency and thrust, with the mean values of $\eta = 0.46$ and $C_T = 2.06$. The highest thrust is generated by cluster F, with mean $C_T = 4.88$ however with a mean efficiency of $\eta = 0.31$.

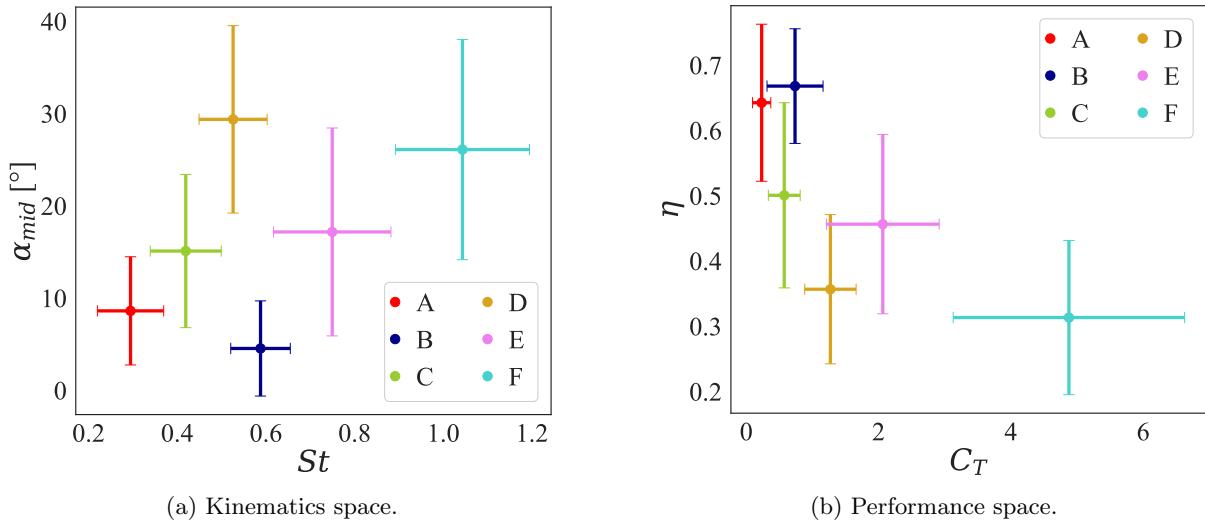


Figure 12: Map of the clusters in the kinematic and performance spaces. The mean and standard deviation of each parameter are separately calculated for each cluster and presented as an error bar plot, demonstrating how the clusters correspond to different regions in both the kinematic and performance spaces.

6. Discussion and Conclusions

The results have shown that the pertinent information with regards to performance and kinematics of an oscillating foil can be correlated with the wake structures through machine learning. This method is demonstrated through a CFD database that includes 121 unique kinematic pitching and heaving strokes that span a range of propulsive regimes at $Re = 10^6$. This dataset specifically targets high-pitch and high-heave modes that produce chaotic vortex wakes, further confounded by the high Reynolds number and thus faster rate of vortex break-up when compared with lower Reynolds number flows.

The methodology utilizes a convolutional autoencoder to distill the information from the wake images into a reduced or latent space. Using Lasso regularization to avoid overfitting and skip connections to solve the vanishing gradient problem, the autoencoder provides a compressed array composed of 10 values which is accurately representative of the original images. The images are then clustered based on the representation in latent space using a k-means++ algorithm.

A one degree-of-freedom model of flow over a pitching foil has three well-defined wake classifications, as previously used by Colvert et al. [9]. Using this model, the autoencoder is tuned, and the clustering demonstrates 100% accuracy even with a low number of samples used for training.

The methodology is then applied to the two degrees-of-freedom model that generates vortex wakes from flow over a pitching and heaving foil. In this model neither the wake classifications nor the number of clustering bins is known. Using the same hyperparameters as the one-degree-of-freedom model, the autoencoder reduces the images to a latent space and the clustering algorithm automatically sorts the images based on their vortex wake profiles. To

assess the results of the methodology, the resulting clusters are mapped to the kinematic space, and to the resulting performance space. These results show that kinematics with similar performance profiles are clustered together, and the propulsive regimes are clustered into similar groups, supporting the conclusions from previous works that the wake can be a predictive measure of underlying kinematics and propulsive performance.

The ability to automatically detect subtle differences in the wake patterns of propulsors will help to develop blended, or physics-informed machine learning models for sensing and control methodologies, particularly important for cooperation among multiple foils in an array configuration. In this case the kinematics and performance of each individual foil is dependent on the wake structure (and thus the kinematics and performance) of those foils upstream. Rather than relying purely on data-driven methodologies, understanding the various classifications and formations within the wake will lead to a better understanding of the underlying flow phenomena for predictive modeling. Although oscillating foils are utilized in this paper, these concepts can extend to other bluff body fluid flows where the wake characteristics can be important for the physical and mathematical modeling of the flow.

While the proposed scheme has proven to work well for the oscillating foil datasets, it has also revealed opportunities for improvement. Due to the interest in the chaotic wakes of high heave amplitudes, there is a low density of data for heave amplitudes (see figure 5b). Thus, it is likely that more importance is given to the heave amplitude in the clustering algorithm, as the width of the wake is highly influenced by this parameter and can be easily deduced by the vortex wake pattern. Including more samples of varying heave amplitudes may force the clustering algorithm to consider the overall vortex pattern independent of the wake width. However the result may be same, as the heave amplitude does play an important role in the Strouhal number and relative angle of attack, both dominant kinematic factors, and thus clustering heavily based on heave amplitude may prove to be a good strategy. Another strategy would be to employ adaptive windows, and thus purposefully focus on vortical patterns and eliminate the response to wake size.

In this process, the autoencoder hyperparameters are tuned for the simple one-degree-of-freedom model, and then successfully applied to the more complex two-degree-of-freedom model without re-tuning. It has not been determined if the tuned autoencoder would work successfully in more diverse vortex wakes (other than pitching and heaving foils), or other flows in general. However the use of convolutional layers and skip-connections adds robustness such that future applications could be easily trained and hyperparameters determined with minimal training data.

7. Acknowledgements

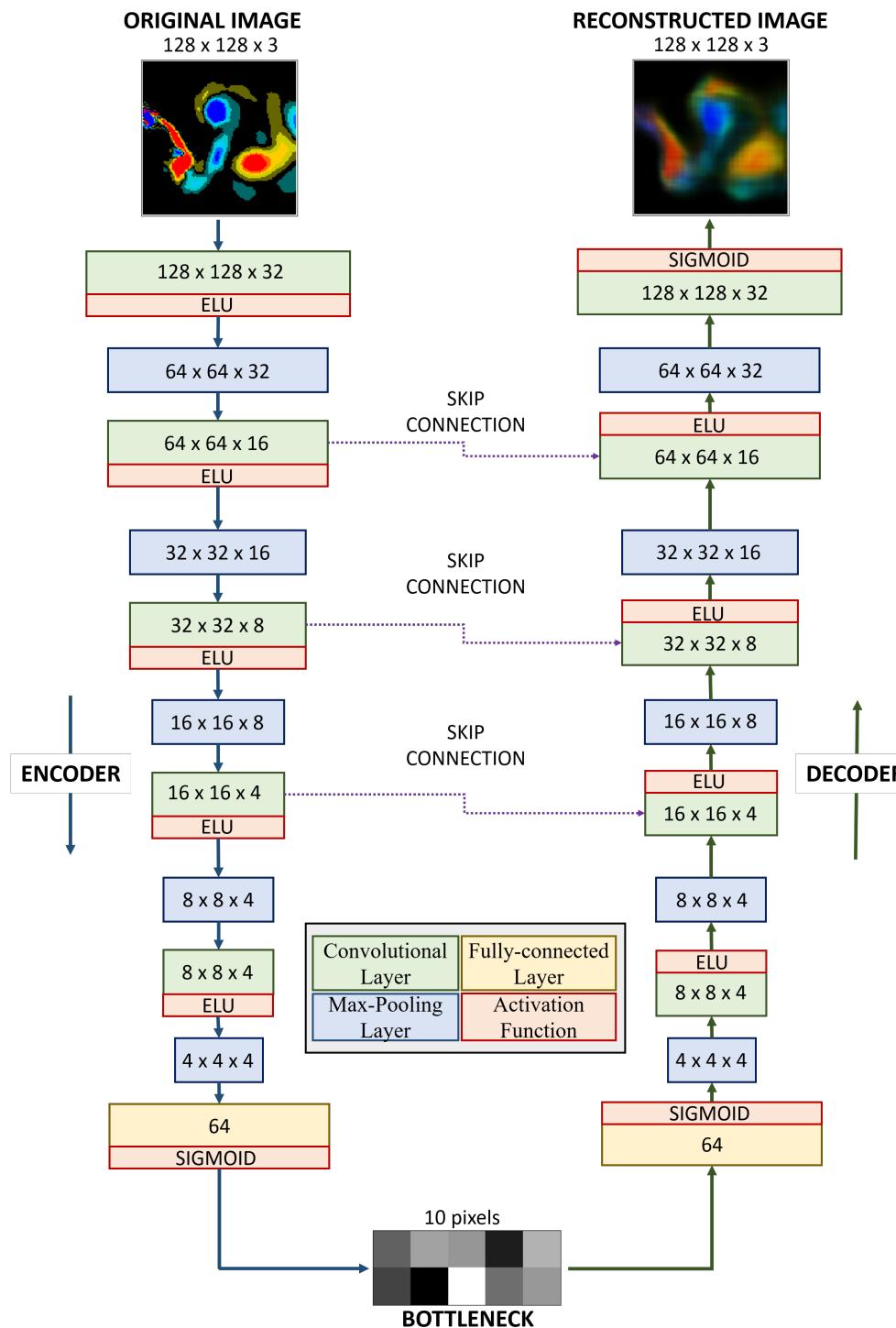
Part of this research was conducted using computational/visualization resources and services at the Center for Computation and Visualization, Brown University. AGC: Methodology, Analysis, Writing - original draft. MD: Simulations, Analysis, Writing - revising/editing. JAF: Conceptualization, Methodology, Writing - revising/editing.

References

- [1] C H K Williamson and A Roshko. Vortex formation in the wake of an oscillating cylinder. *J. Fluids Struct.*, 2(4):355–381, jul 1988.
- [2] Teis Schnipper, Anders Andersen, and Tomas Bohr. Vortex wakes of a flapping foil. *J. Fluid Mech.*, 633:411–423, 2009.
- [3] Manoochehr M Koochesfahani. Vortical patterns in the wake of an oscillating airfoil. *AIAA J.*, 27(9):1200–1205, sep 1989.
- [4] J C S Lai and M F Platzer. Jet characteristics of a plunging airfoil. *AIAA J.*, 37(12):1529–1537, jan 1999.
- [5] Z Jane Wang. Vortex shedding and frequency selection in flapping flight. *J. Fluid Mech.*, 410:323–341, may 2000.
- [6] F S Hover, Ø Haugsdal, and M S Triantafyllou. Effect of angle of attack profiles in flapping foil propulsion. *J. Fluids Struct.*, 19(1):37–47, 2004.
- [7] Qing Xiao and Wei Liao. Numerical investigation of angle of attack profile on propulsion performance of an oscillating foil. *Comput. Fluids*, 39(8):1366–1380, sep 2010.
- [8] Daniel Floryan, Tyler Van Buren, and Alexander J. Smits. Swimmers' wake structures are not reliable indicators of swimming performance. *Bioinspiration and Biomimetics*, 15(2):024001, feb 2020.
- [9] Brendan Colvert, Mohamad Alsalmi, and Eva Kanso. Classifying vortex wakes using neural networks. *Bioinspiration and Biomimetics*, 13(2), 2018.
- [10] Mengying Wang and Maziar S. Hemati. Detecting exotic wakes with hydrodynamic sensors. *Theor. Comput. Fluid Dyn.*, 33(3-4):235–254, aug 2019.
- [11] Binglin Li, Xiang Zhang, and Xing Zhang. Classifying wakes produced by self-propelled fish-like swimmers using neural networks. *Theor. Appl. Mech. Lett.*, 10(3):149–154, mar 2020.
- [12] Bernardo Luiz R Ribeiro, Sarah L Frank, and Jennifer A Franck. Vortex dynamics and Reynolds number effects of an oscillating hydrofoil in energy harvesting mode. *J. Fluids Struct.*, 94:102888, apr 2020.
- [13] David Arthur and Sergei Vassilvitskii. k-means++: The Advantages of Careful Seeding. Technical Report 2006-13, Stanford InfoLab, jun 2006.
- [14] Mukul Dave, Arianne Spaulding, and Jennifer A Franck. Variable thrust and high efficiency propulsion with oscillating foils at high Reynolds numbers. *Ocean Eng.*, 214:107833, oct 2020.

- [15] Richard P Dwight. Robust Mesh Deformation using the Linear Elasticity Equations. In *Comput. Fluid Dyn. 2006*, pages 401–406. Springer Berlin Heidelberg, 2009.
- [16] Frank Martijn Bos. *Numerical simulations of flapping foil and wing aerodynamics: Mesh deformation using radial basis functions*. PhD thesis, TU Delft, 2010.
- [17] Ravi Ramamurti and William Sandberg. Simulation of flow about flapping airfoils using finite element incompressible flow solver. *AIAA J.*, 39(2):253–260, jan 2001.
- [18] Lionel Schouveiler, F S Hover, and M S Triantafyllou. Performance of flapping foil propulsion. *J. Fluids Struct.*, 20(7):949–959, oct 2005.
- [19] Alexandra H Techet. Propulsive performance of biologically inspired flapping foils at high Reynolds numbers. *J. Exp. Biol.*, 211(2):274–279, 2007.
- [20] Daniel Floryan, Tyler Van Buren, Alexander J. Smits, Tyler Van Buren, and Alexander J. Smits. Large-amplitude oscillations of foils for efficient propulsion. *Phys. Rev. Fluids*, 4(9):093102, sep 2019.
- [21] A Alqahtani, X Xie, J Deng, and M W Jones. A Deep Convolutional Auto-Encoder with Embedded Clustering. In *Proc. - Int. Conf. Image Process. ICIP*, pages 4058–4062. IEEE Computer Society, aug 2018.
- [22] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [23] François Chollet et al. Keras, 2015.
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F Pereira, C J C Burges, L Bottou, and K Q Weinberger, editors, *Adv. Neural Inf. Process. Syst. 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [25] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, volume 8689 LNCS, pages 818–833. Springer Verlag, 2014.
- [26] G E Nasr, E A Badr, and C Joun. Cross Entropy Error Function in Neural Networks: Forecasting Gasoline Demand. In *FLAIRS-02*, pages 381–384, 2002.
- [27] Zhilu Zhang and Mert R Sabuncu. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In *32nd Conf. Neural Inf. Process. Syst.*, pages 8778–8788, Montréal, Canada, 2018.
- [28] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* International Conference on Learning Representations, ICLR, dec 2015.
- [29] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. In *30th Conf. Neural Inf. Process. Syst. (NIPS 2016)*, pages 2802–2810, Barcelona, Spain, 2016.
- [30] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, 6(2):107–116, nov 1998.
- [31] Xin Jin and Jiawei Han. *K-Medoids Clustering*, pages 564–565. Springer US, Boston, MA, 2010.
- [32] Sakshi Patel, Shivani Sihmar, and Aman Jatain. A study of hierarchical clustering algorithms. In *2015 2nd Int. Conf. Comput. Sustain. Glob. Dev.*, pages 537–541, mar 2015.
- [33] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD-96*, pages 226–231, 1996.
- [34] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, jan 2012.
- [35] Stuart P Lloyd. Least Squares Quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–137, 1982.
- [36] Robert L Thorndike. Who Belongs in a Family?, 1953.
- [37] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20:53–65, 1987.

Appendix A. Convolutional Autoencoder Architecture



Tuned convolutional autoencoder architecture.