Optimizing Wind Turbine Placement Subject to Turbine Wakes

Yu Shen Lu
yushenlu@stanford.edu

Manasi Sharma
manasis@stanford.edu

Peiyun Zhu
zhpyun@stanford.edu

Abstract

Wind energy is an important to help reduce CO_2 emission and combat climate change. However, placement of turbines in wind farm is a difficult decision which affects the productivity and efficiency of the energy generation. In our project, we aim to optimize wind turbine placement in a wind farm subject to the effect of wakes, which is one of the main sources of uncertainty in wind farm power extraction. We model this problem as a Markov Decision Process using Q-learning for a small grid, and then generalize to large grids using neural network. We also analyze the effect of adding cost to install turbines and the effect of different parameters of the wake model.

1 Introduction

Wind is one of the most fast growing renewable energy resources[1], which has been playing an increasingly important role in providing the world clean and sustainable energy. Extracting energy from wind turbines, however, faces technical challenges from many aspects. One critical decision that needs to be made when building a wind farm is the placement of turbines. There are two main sources of uncertainty in the environmental state that strongly affect the efficiency of the energy extraction of the wind farm as a whole. First is the wind direction, which is dynamic and distributed cross all angles depending on the local meteorological condition. Wind turbines extract the maximum amount of energy when the wind direction is perpendicular to the plane that the rotor blades span, but this cannot be satisfied all the time or for every individual turbine. Second is the turbine wakes, which refers to the velocity field behind the turbines. Because the flow field is disturbed by the turbine blades and energy is extracted, the wake is highly turbulent with much reduced wind speed. In wind farms with multiple turbines, the placement of turbines can be optimized to reach the maximum efficiency of energy extraction.

2 Problem formulation and related works

Our project is to optimize wind turbine placement in a wind farm subject to the effect of wakes. Some previous studies used multi-agent deep reinforcement to optimize wind farm performance [2], which optimizes fatigue damage by controling the pitch and yaw of all turbines. They reduces the dimensionality by an auto-encoder and used domain knowledge to prune the action space. Other studies focused on optimizing wind turbine placement using mathematical optimization methods (paper in preparation), or Bayesian Optimized Monte Carlo Planning to optimize sensor tower locations in a wind farm to generate accurate maps of high wind areas. They model the problem as a partially observable Markov decision process (POMDP) [3], which requires an observation model to account for the measurement uncertainty of wind speed.

As a first-order approach, we do not consider the state uncertainty and formulate the turbine placement problem as a Markov Decision Processes (MDP) by placing each turbine sequentially. This is because we can simplify the problem by accounting for the uncertainty due to turbine wakes

Preprint. Under review.

using a wake model, described in section 2.1. In this way, we believe that our approach is simpler than the POMDP approach and can be applied in limited areas where the wind patterns are fairly static and well-known. We illustrate our method in section 3, and discuss the influence of wake model parameters and different reward formulation on the final results. Finally conclusion and future work is discussed in section 6.

2.1 Wake model

We obtain wind speed data from Global Wind Atlas [4]. The location coordinates are equally spaced longitude and latitude. We converted them to meters, resulting in a spacing of approximately 200 m in x and 350 m in y. The data set contains the wind velocity u at 100 meter high on a 96×102 horizontal grid. In the model design, we try to capture two environmental uncertainties. First is the wind turbine wake, and second is the varying wind direction. Numerous turbine wake models exist in literature, most of which are highly parameterized and idealized. To account for the uncertainty in the actual wake values, we use a Gaussian distribution to generate wake, with the mean value computed by the Risoe WAsP model [5]. The Risoe WAsP model is formulated as

$$U_{wake} = U_{freestream} \left[1 - (1 - \sqrt{1 - C_T}) \left(\frac{D}{D + 2k_{wake}x}\right)\right] \tag{1}$$

where U_{wake}, $U_{freestream}$ are the wake and free stream velocity, C_T is the thrust coefficient of the wind turbine, D is the rotor diameter, k_wake is the wake decay constant, and x is the down stream distance to the turbine. For onshore wind farms, $k_{wake} = 0.075$ [5]. Under optimal conditions, C_T can be one, so we use $C_T = 1$ [6]. The average rotor diameter (D) is 125 m. To compute the wake after a turbine, we use the current wind speed at this turbine location to substitute $U_{freestream}$. The value computed by eq.(1) will be the mean of the Gaussian distribution. Turbulent intensity is high in the wake region [7], and the angle between the direction where the turbines align and the wind direction also leads to different wake deficit. The power loss in the wake can be 1% to 50% of the power of the turbine, depending on different wind directions [8].These factors makes the wake values very uncertain and cannot be approximated by an deterministic equation. To resemble the range of variation caused by turbulence and wind direction, we set 30% of the mean to be the standard deviation of the Gaussion distribution [8].

We implemented the wake function as part of our MDP problem. According to [7], the near wake is around 2-4 D after the turbine, so in our model, if a newly added turbine is within $5D$ Euclidean distance downstream an existing turbine, it is influenced by the wake and we will calculate the wake velocity. The new turbine can be in the wake region of several different upstream turbines. To simplify the problem, we compute wakes generated by upstream turbines one by one, and take the highest velocity loss to be the wake at the new turbine location.

The reward of this MDP problem is the power extracted by the newly added turbine, which is computed by

$$P = \frac{1}{2}\frac{\pi}{4}D^2\rho u^3 \tag{2}$$

[9], where ρ is the air density, and u is the wind speed at the turbine hub height.

3 Method

We start approaching this problem by first narrowing down the state space. randomly sampled a 3x3 grid in the wind atlas and restricted our agent to this space. This is equivalent to a 600m $\times$ 1050m area. Our state space is a binary mask of if a turbine exist at the grid location, which means we have $2^{3\times3} = 512$ states in total. The transition of our model is entirely deterministic, since stochastic transition fails to capture the problem realistically (i.e. people usually do not accidentally place turbine at unwanted locations). Notice we do not give the model any information of the actual wind or the wake effect, the policy needs to learn about these through experiences. We later increased the size of the grid and used a Neural Network to predict the utility and extract the policy. We decided to use Q-learning as we felt a model-free approach would be better suited to this problem because of the large state space (to avoid a substantial number of parameters).

An action is to choose a grid point to place a turbine, or to stop adding new turbines. Thus the action space is the size of the grid of the wind-farm, plus one extra action to stop adding turbine.

Every action except the stop action corresponds to the location at which to add the turbine. Our agent must make decisions sequentially as the previously placed turbine will decrease wind-speed for turbines that are in the area affected by turbine wake, making this a sequential decision problem. The agent also needs to decide when to stop placing additional turbine without placing additional turbines that only bring marginal reward.

At each step, the reward of our model is calculated by approximating the additional power generated by the placement of a turbine, as seen in equation 2. To ensure that our agent does not learn to produce invalid actions (such as placing a turbine on top of an existing one), we give our agent an negative reward equivalent to -1 MW whenever the action is invalid.

We began with a random policy, to generate a dataset of size 100,000 for offline Q learning. Utility is then calculated with the update function, and policy is extracted greedily with respect to the utility values. Later, as we increase our grid size, our state space grows exponentially, and we encounter unseen states when creating a policy. We approximate unseen Q-values using a trained Neural Network. We also closely examined the effect of changing various parameters in wake and reward functions on the final policy.

4 Results

4.1 Q learning

We found that for an wake region of 5D distance behind the turbin and $\sigma = 0.1\mu$ (discussed later), the optimal policy for adding turbines was at the following locations and in the following order: $(0, 2) \rightarrow (0, 0) \rightarrow (0, 1) \rightarrow (2, 1) \rightarrow (2, 2) \rightarrow (1, 0) \rightarrow (1, 2) \rightarrow (1, 1) \rightarrow (2, 0)$

This is visualized in Figure 1, where the blue points depict the locations at which turbines are added and in what order.

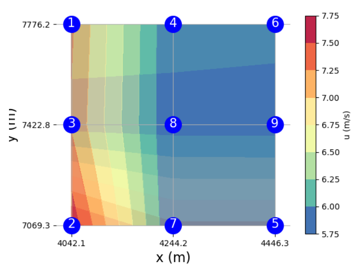


Figure 1: The optimal wind turbine layout for a 3X3 grid with a 5D wake extent and $\sigma = 0.1\mu$

We see that a turbine is placed at every location. Since we have not included the cost of the turbine in the reward yet (discussed later), the policy increases the power by placing a turbine at every point (even when the resulting power from a turbine may be greatly reduced due to wake effects of other turbines). However, it is important to note that the earlier turbines 1-3, which would be prioritized if we were to limit the maximum number of turbines, are placed in locations with highest wind velocity and vertically to avoid wake effects.

The total power extracted from the grid is $6.524359 MW$, as compared to a random policy, which results in a power of $3.597243 MW$. Our Q-learning approach almost doubles the amount of power that can be extracted from the grid with an optimal placement of turbines.

4.2 Q learning with neural network

Since the number of states grows exponentially with the size of the grid, and because we had to know which states are specifically unvisited to be able to update the corresponding Q-values, we decided to limit the number of turbines to be placed for larger grids. We limited the number of turbines to be a maximum of 5. This means our agent was forced to produce a stop action after 5 steps.

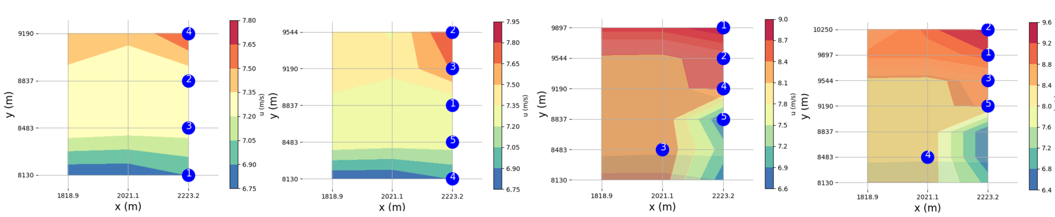
The Neural Network we used was a simple one layer Multi-layer perceptron with hidden layer size of 4, for which we used a Keras implementation [10].

We plotted the total power from the optimum policy returned by Q-learning (which was run with the CNN from grid 3x5 onward), compared with the power outputs for a random policy, to a grid size up to 3x7. We chose to increase the grid only in the vertical (y) direction as the wake is only in the x-direction. The number of states visited using a random exploration policy, the number of unvisited states whose Q-values are predicted using a CNN, the total power generated from the optimized policy placement and corresponding random policy placement are summarized in Table 1 below.

		Sequence turbines added in	Num states visited	Num states unvisited	Total power from optimal policy (in MW)	Total from random policy (in MW)
Wind map grid size	3x4	(2, 0), (2, 2), (2, 1), (2, 3)	1586	0	11.5675	4.5743
	3x5	(2, 2), (2, 4), (2, 3), (2, 0), (2, 1)	4930	14	15.1725	5.0592
	3x6	(2, 5), (2, 4), (1, 1), (2, 3), (2, 2)	11,536	1080	15.4187	5.7391
	3x7	(2, 5), (2, 6), (2, 4), (1, 1), (2, 3)	19,160	8736	21.3124	6.1333

Table 1: Statistics for the increasing grid sizes

The turbine plots and their effect on the wind map are also shown for each of the increasing grid sizes in Figure 2.



(a) 3x4 grid (b) 3x5 grid (c) 3x6 grid (d) 3x7 grid

Figure 2: The optimal wind turbine placement using $5D$ as the wake extent, and different standard deviation of the wake model, overlying on the contour of wind speed. Turbines are marked as blue dots on the grid, and the white number labels indicate the sequence when the turbines are added.

From Figure 2, we see that when we limit the maximum number of turbines for larger grids, the optimal policy seems to be to space the wind turbines vertically (as seen in grids 3x4 and 3x5). This makes intuitive sense as the wake effects of turbines are in the x direction, and thus such a vertical placement enables minimum wake interference among turbines. The optimal policy also takes into consideration the wind speed of the wind map at that particular instance, as seen in grid 3x6, for which the third turbine is placed at location (1,1) instead of (2,1) as the wind speed, as thus energy extracted, is higher at (1,1); this goes for 3x7 as well, where we can see that turbine number 4 is placed at (1,1) instead of (2,1). Note: since we are limiting the number of turbines, the proximity of the turbines in the y-direction does not matter in this case as the wake effects are only in the x-direction.

In addition, we see that the CNN seems to perform relatively well in preserving the core of the optimal policies from grids 3x4-3x5 (with minimal unvisited states), still lining up. However from

Table 1, we can see that the advantage of the optimal policy compared to the random policy for that grid decreases as the grid size increases. This can be attributed to the uncertainty and possible bias in utilizing a CNN to predict Q values for unvisited states, that reduces the optimality of the returned policy and thus its utility compared to the random policy.

5 Discussion

5.1 The influence of the wake model

As mentioned in Section 2.1, we use a Gaussian distribution to model the wake to account for uncertainties due to the form of the wake model, turbulence and wind direction. We set the mean of the Gaussian distribution, μ, by eq.(1), but the standard deviation, σ is set heuristically. The default value we use for σ is 0.3μ, but it can vary depending on the uncertainty of the actual wake. In our model, we specify that the wake region of a turbine is within $5D$ distance behind the turbine, but this extent is empirical and could vary case by case as well. In this section, we experiment with different values for σ and the extent of wake influence, to show how wake of different strength can influence the optimal wind turbine layout that the Q-learning algorithm determines. We chose a 3×3 sub-grid of the wind map (starting from index 20 in both x and y axes), and use the same sub-grid to compare different parameter values of the wake model.

5.1.1 The influence of wake standard deviation

The original wind map of the sub-grid is shown in Fig.3a. We tried five values of the wake model standard deviation from 0.1μ to 1.0μ, and present the optimum wind turbine layout in Fig.3, which overlies on the wind speed contour. The turbines are also labeled by the sequence they were added. Note that there are only nine points in the grid, and the wind speed is essentially discrete. The color filling of the contour is the result of linear interpolation between the grid points. The wake extent is fixed to be $5D$.

The wake effect significantly changes the wind speed of the sub-grid. Behind the first row of turbines (where $x = 4042.1$ m), velocities are zero or near zero, and it is the first row of turbines that extract the most energy. Our algorithm is successful because it learns to place turbine where velocities are higher. For example, in Fig.3c, it first placed a turbine in the last row, and then learned to place the second and the third in regions where wake does not influence the first turbine. Then it placed another two turbines at where the velocity is non-zero, and then stopped.

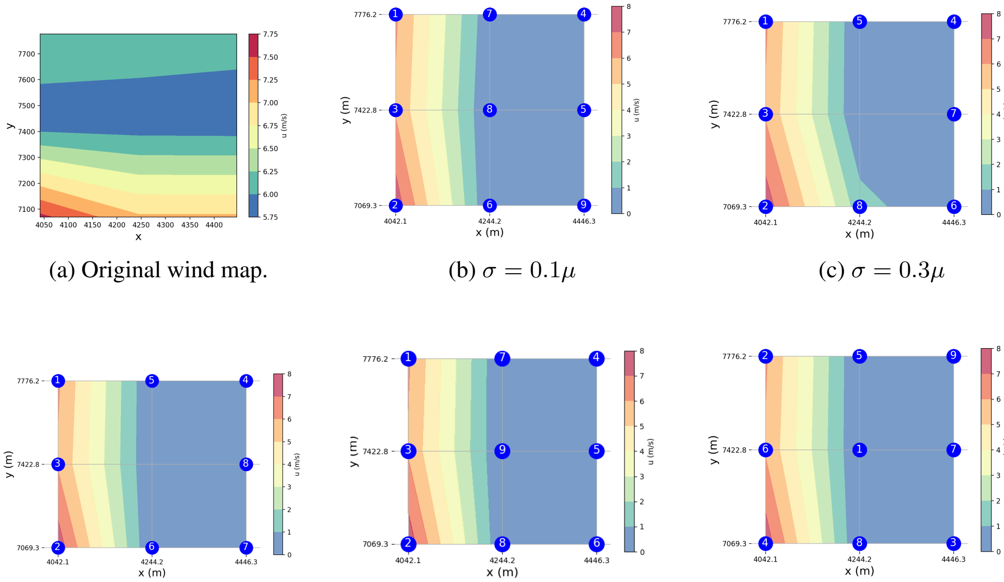
When we decrease the standard deviation, the wake tends to zero out the velocity after the first row, because a narrow Gaussian distribution tends to sample values close to the mean wind speed at the turbine, μ, which is large in the first row. That results in stronger wake and less total power, as shown in Table.2 and Figs.3b and 3c.

For higher standard deviation, such as $\sigma = 0.5\mu$, 0.6μ or 1.0μ, the wake can have extremely large values, so it could zero out velocity as well. The resulting power are similar between these cases because they end up in very similar final wind maps.

Note that in Figs.3d to 3f, the algorithm placed turbines at grid points with zero velocity too. This is because we did not include a cost of installing a turbine in this section, so even if the immediate reward at these points is zero, it still does not hurt to place one there. In reality, however, the optimal layout is only to place turbines in places with non-zero velocity.

σ	0.1μ	0.3μ	0.5μ	0.6μ	1.0μ
Total power (MW)	6.524484	6.545986	6.524359	6.524362	6.525013

Table 2: Total power extracted of the final layout using different σ values in the wake model.



(a) Original wind map. (b) $\sigma = 0.1\mu$ (c) $\sigma = 0.3\mu$

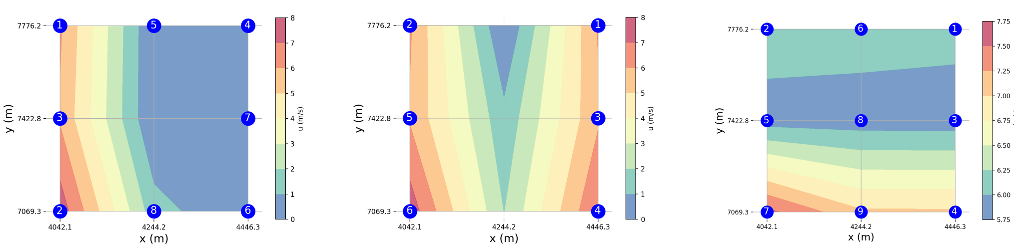
(d) $\sigma = 0.5\mu$ (e) $\sigma = 0.6\mu$ (f) $\sigma = 1.0\mu$

Figure 3: The optimal wind turbine layout using $5D$ as the wake extent, and different standard deviation of the wake model, overlying on the contour of wind speed. Turbines are marked as blue dots on the grid, and the white number labels indicate the sequence when the turbines are added.

5.1.2 The influence of wake extent

Using the same sub-grid, we fix the standard deviation to be 0.3μ and tried different values for the wake extent: $5D$, $2D$ and $1D$. $5D = 625$ m while the grid spacing in x is 200 m. In a 3×3 grid, every point is within the wake region of any other point. If the extent shrinks to $2D = 250$ m, then only points next to each other will be influenced by wake, e.g. if there are only turbines in the first row, then only the second row experiences wake while the wind speed in the third row is not affected. If the extent is $1D = 125$ m, then there will be no wake effect wherever the turbine is.

The results are expected, as shown in Fig.4. For an extent of $5D$, both only the first and third rows are occupied by turbines to minimize the effect of wake; for $2D$, the velocity in the third row is high because it is not influenced by the turbines in the first row, and the algorithm learned to not to place turbines in the second row to maximize power extraction; for $1D$, the wind map is the same as when there is no turbine because there is no wake effect, and the optimal strategy is simply to put turbines everywhere.

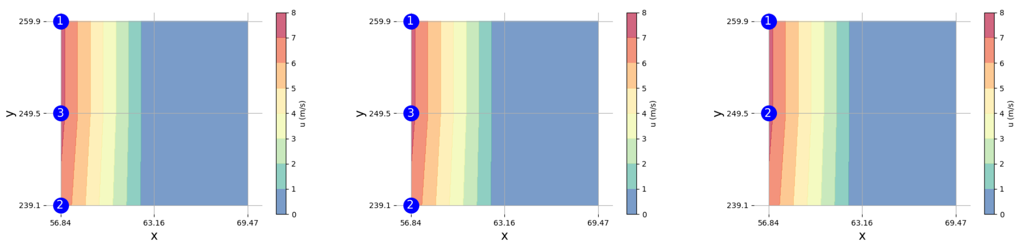


(a) $5D$ extent (b) $2D$ extent (c) $1D$ extent

Figure 4: The optimal wind turbine layout using 0.3μ as the standard deviation of the wake model, and different wake extent, overlying on the contour of wind speed. Turbines are marked as blue dots on the grid, and the white number labels indicate the sequence when the turbines are added.

5.2 Influence of turbine cost

So far in the experiment we allow agent to build turbine with no cost, and the reward function is purely based on the power generated and the validity of the action. This results in the behavior where it try to build turbine at all possible locations even if some locations yield no power production at all, as shown in the previous sections. Sometimes, due to the stochastic nature of the wake effect, placing wind turbine in regions affected by wake can yield a small amount of power generated, as a result, our policy tend to be overly optimistic and try to capture all the possible reward by placing more turbines than reasonable. However, in reality, building turbine will incur an significant cost to the builder, which we can simulate by adding a flat negative reward to the agent whenever it builds a new turbine.



(a) turbine cost = $0.5kW$ (b) turbine cost = $50kW$ (c) turbine cost = $2.5MW$

Figure 5: The policy quickly stabilize to 3 turbine (compare to 8 9 turbines in Figure 3) with cost added, this policy does not change until cost is increased to dominate over possible reward from building turbines.

Surprisingly, even a small cost to building turbine can stabilize the policy and most of different costs we tried converges to the same policy – placing three turbine in the same column in the order of the local wind speed. Placing a new turbine in area free of wake can yield power generation of around 2.3 $2.9MW$, but event at cost of $-0.5kW$, the policy quickly get rid of any turbines that are placed inside area with wake. This effect is identical for all the values of cost tested from $0.5 - 2000kW$, and as we increase the cost of turbine to be higher than the power generated, the greedy policy decide to stop prematurely as shown in Fig 5c. When cost of turbine is equivalent to $3MW$ generated, the policy simply stops for all states.

6 Conclusion and Future work

We formulate the wind turbine placement problem as an MDP, and used Q learning to estimate the utility of each state, i.e. each possible layout of the turbines. We start with a small grid, and then generalize to larger state spaces by a neural network. The results show that our algorithm significantly outperforms a random policy.

The uncertainty of the environment was captured by our wake model, which considers the effect of turbulence of wind direction distribution. One parameter, the wake extent has a large impact on the optimal solution, indicating the importance of wake in wind farm planning; another parameter, the standard deviation of the Gaussian wake model controls the strength of the wake but have smaller impact. Adding a cost for installing a turbine improved the optimal solution by not wasting effort to place turbines at where there is negligible wind velocity.

Despite the success of our algorithm, our wake model is very simplified and the Gaussian formulation may deviate from reality. Refining the wake model would improve the performance for more realistic problems. Additionally, we did not consider any uncertainty of the observed wind speed. It would be beneficial to include uncertainty in observations in the future work and use POMDP approach to solve this problem, which brings it closer to reality.

References

[1] Enas Taha Sayed et al. “A critical review on environmental impacts of renewable energy systems and mitigation strategies: Wind, hydro, biomass and geothermal”. In: *Science of*

The Total Environment 766 (2021), p. 144505. ISSN: 0048-9697. DOI: https://doi.org/10.1016/j.scitotenv.2020.144505. URL: https://www.sciencedirect.com/science/article/pii/S0048969720380360.

[2] Venkata Ramakrishna Padullaparthi et al. "FALCON-FArm Level CONtrol for wind turbines using multi-agent deep reinforcement learning". In: *Renewable Energy* (2021).

[3] John Mern et al. "Bayesian Optimized Monte Carlo Planning". In: *CoRR* abs/2010.03597 (2020). arXiv: 2010.03597. URL: https://arxiv.org/abs/2010.03597.

[4] "The Global Wind Atlas". In: (). URL: https://globalwindatlas.info/.

[5] R J Barthelmie et al. "Comparison of wake model simulations with offshore wind turbine wake profiles measured by sodar". In: *Journal of atmospheric and oceanic technology* 23.7 (2006), pp. 888–901.

[6] Marshall L Buhl Jr. *New empirical relationship between thrust coefficient and induction factor for the turbulent windmill state*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2005.

[7] Tuhfe Göçmen et al. "Wind turbine wake models developed at the technical university of Denmark: A review". In: *Renewable and Sustainable Energy Reviews* 60 (2016), pp. 752–769.

[8] M Gaumond et al. "Evaluation of the wind direction uncertainty and its impact on wake modeling at the Horns Rev offshore wind farm". In: *Wind Energy* 17.8 (2014), pp. 1169–1178.

[9] Di Yang, Charles Meneveau, and Lian Shen. "Effect of downwind swells on offshore wind energy harvesting–a large-eddy simulation study". In: *Renewable Energy* 70 (2014), pp. 11–23.

[10] Francois Chollet et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.