Intelligent Wind Farm Control via Grouping-Based Reinforcement Learning

Hongyang Dong and Xiaowei Zhao

Abstract— **This paper aims to maximize the total power generation for wind farms subject to strong wake effects and stochastic inflow wind speeds. A data-driven control method that only requires the accessible measurements of every turbine in the farm is proposed via deep reinforcement learning (DRL). We employ a grouping strategy to mitigate the high computational complexity induced by DRL and enhance our method's applicability to large-scale wind farms. Based on the levels of aerodynamic interactions among turbines, this grouping strategy divides the whole farm into small sub-groups. Therefore, one can execute DRL on these sub-groups instead of carrying on a complicated learning process for the entire farm. Simulations verify the advantages of the proposed DRL-based wind farm control method over the commonly employed greedy strategy. Results also show that the proposed method can significantly reduce the overall computing cost compared with the direct execution of DRL on the whole wind farm.**

I. INTRODUCTION

Wind energy is one of the most efficient green energy, and it is essential for the global goal towards zero-carbon emissions. Currently, 743 GW wind energy capacity has been installed worldwide, helping decrease over 1.1 billion tons of CO2 [1]. In particular, 15 new offshore wind farms were put into operation in 2020, and 30 more are currently under construction. With the rapid development of wind farms, maximizing wind farms' operating efficiency has become an essential topic that has received great attention from both industry and academia. As reported by many studies [2], [3], [4], [5], the farm-level power generation can be significantly influenced by the wake effects among turbines [6], [7]. Under this context, the traditional greedy strategy (in which every single turbine in the farm only cares about maximizing its own power generation) can result in reduced farm-level power generation efficiency, e.g., wake effects lead to a 20% annual generation loss of Denmark Horns Rev Offshore Wind Farm. Therefore, many recent studies, e.g. [3], [4], [5], [8], [9], [10], explore to control all turbines in the farm cooperatively to mitigate wake effects and increase the whole farm's economic profitability. For example, site tests in [4] verified that properly controlling every turbine in the farm can steer wakes and potentially increase the farm-level power generation.

Most of the existing methods to maximize farm-level power generation are optimization-based. They first employ/build analytical models for wakes or wind farm simulators to map the relationship between wind farm states (e.g., induction factors and yaw angles of all turbines) and the farm-level power generation. After that, different optimization methods can be developed to decide settings or control inputs for every turbine in the farm. Based on the famous Park model, Ref. [9] searched the optimal induction factors for a three-turbine wind farm via a game-theoretic (GT) method. Similarly, Ref. [10] utilized the GT method to optimize turbine yaw settings based on a parametric model developed by the authors. A sequential convex programming strategy was employed in [11] and applied to a model tweaked from the Jensen wake model [12], and a Bayesian-based optimization method was designed in [13] for the same purpose. All these elegant results show that cooperatively operating all turbines in the farm can improve the farm's power capture efficiency and increase the overall power generation.

However, these optimization-based methods have several limitations. They usually require steady-style data to carry out searching/learning. Such data typically come from steady-state wind farm models, which have limited fidelity and inevitable modelling errors. Employing data from more practical environment conditions (e.g. under time-varying wind speeds), on the other hand, can make these optimization-based methods unstable. Therefore, unmatched or degraded performance may be observed in practical applications of optimization-based methods. In addition, these methods only can provide unchanged settings for turbine states (e.g., induction factors and yaw angles). They cannot achieve closed-loop control based on real-time external/internal conditions, rendering their performance sub-optimal and limited. Aiming to address these issues, Refs. [8], [14] developed model predictive control methods to adjust turbines' induction factors in real-time. Their case studies verified that closed-loop control could lead to clearly better performance than quasi-steady-state optimization. However, a drawback of these important results is that the full states at all spatial cells of the staggered grid (which is used to discrete the flow field) are required by the controller. But such information may be hard to measure for real wind farms.

These facts indicate that innovative technologies are required to achieve closed-loop wind farm control with measurable states under dynamic environmental conditions. Deep reinforcement learning (DRL) is a promising choice to handle this challenging task. DRL is a state-of-the-art AI and data-driven control technology. The core advantage of

This work was funded by the UK Engineering and Physical Sciences Research Council (grant numbers: EP/R007470/1, EP/S000747/1).

H. Dong and X. Zhao (Corresponding Author) are with the Intelligent Control and Smart Energy (ICSE) Group, School of Engineering, University of Warwick, Coventry, CV4 7AL, UK. Emails: hongyang.dong@warwick.ac.uk; xiaowei.zhao@warwick.ac.uk

it is the ability to improve control policies by interacting with environments. It has attracted worldwide research interest and been applied to many important fields [15], [16], [17]. Notably, several recent studies [18], [19], [20], [21] successfully applied DRL to address wind farm control tasks. They verified the feasibility of employing DRL to maximize wind farms' economic profitability. These data-driven results release the requirement of analytical wind farm models, showing strong adaptability and robustness. However, a drawback of DRL-based wind farm control methods is the heavy computational complexity. DRL is based on trial-and-error. It typically requires a relatively large set of data to carry out training and improve its performance gradually, and the complexity grows exponentially as the number of turbines in the farm increases. Reducing computational complexity is of great importance for the practical application of DRL-based wind farm control methods.

Motivated by these facts, a grouping-based DRL method is proposed in this paper to maximize the total power generation for wind farms subject to strong wake effects and stochastic inflow wind speeds. Based on the levels of aerodynamic interactions among turbines, we propose a grouping strategy to divide the whole farm into small sub-groups. After that, we execute DRL on these sub-groups instead of carrying on a complicated learning process for the entire farm. Our method takes advantage of both model-based and model-free wind farm control approaches. On the one hand, it employs wind farm models/simulators to carry out grouping and conduct pre-training for DRL, mitigating the high computational complexity, increasing the learning efficiency, and enhancing our method's applicability to large-scale wind farms. On the other hand, it inherits the core features of model-free DRL methods. After the pre-training, it can employ accessible data (e.g. states and measurements of turbines) to improve the performance of closed-loop wind farm control while without relying on any analytical models, rendering strong adaptability and robustness. Simulations with WFSim [22] verify the advantages of the proposed DRL-based wind farm control method over the commonly employed greedy strategy. Results also show that our method can significantly reduce the overall computational complexity compared with the direct execution of DRL on the whole wind farm.

The remainder of this paper is as follows. First, we formalize the wind farm power maximization problem in Sec. II. Then we explain in detail how we carry out intelligent wind farm control via a grouping-based DRL method in Sec. III. Simulation results with WFsim are provided in Sec. IV. Finally, we conclude this paper in Sec. V.

II. Problem Formulation

We denote a wind farm by $\mathcal{WF}$ and the turbines in it by $\mathcal{WT}_1$, $\mathcal{WT}_2$, ..., $\mathcal{WT}_n$, with n to be the total turbine number. Then we describe the power generation of $\mathcal{WT}_i$ (denoted by E_i), $i = 1, 2, ..., n$, as follows:

$$E_i = H_i(U_i, \alpha_i, \beta_i) \tag{1}$$

where U_i is the inflow wind speed at turbine rotor, α_i is the induction state (such as the induction factor or some other states that are related to the induction factor, e.g. the modified thrust coefficient), and β_i is the yaw angle. Here the function H_i is the mapping from U_i, α_i, β_i to E_i. Therefore, the whole farm's power generation is

$$E = \sum_{i=1}^{n} E_i = \sum_{i=1}^{n} H_i(U_i, \alpha_i, \beta_i) \tag{2}$$

Our goal is to maximize E by controlling α_i, β_i for every turbine in the farm. As mentioned in the introduction, a grouping-based DRL method is developed in the next section to achieve this goal. To make the whole design process easy-to-follow, we consider a case study with a 16-turbine wind farm as shown in Fig. 1. The related simulation is based on the dynamic wind farm simulator (WFSim) developed in [22]. From Fig. 1, one can see that the wakes caused by upstream turbines have a clear impact on downstream turbines. It should be emphasized that the proposed method can be applied to different wind farms, and the one in Fig. 1 is just a prototypical example.

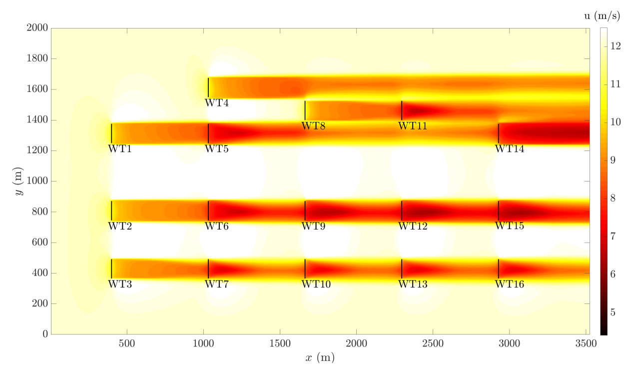

Fig. 1: Illustration of a 16-turbine wind farm.

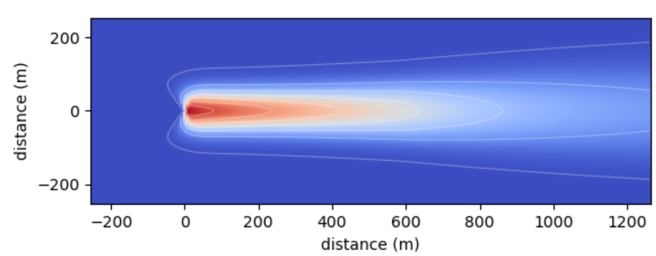

Fig. 2: Illustration of the influence field of a single turbine.

III. Intelligent Wind Farm Control via Grouping-based DRL

In this section, we proposed a grouping-based DRL method to maximize the farm-level power generation (i.e. E) by controlling yaw angles and induction states (i.e. α_i and β_i, $i = 1, 2, 3..., n$).

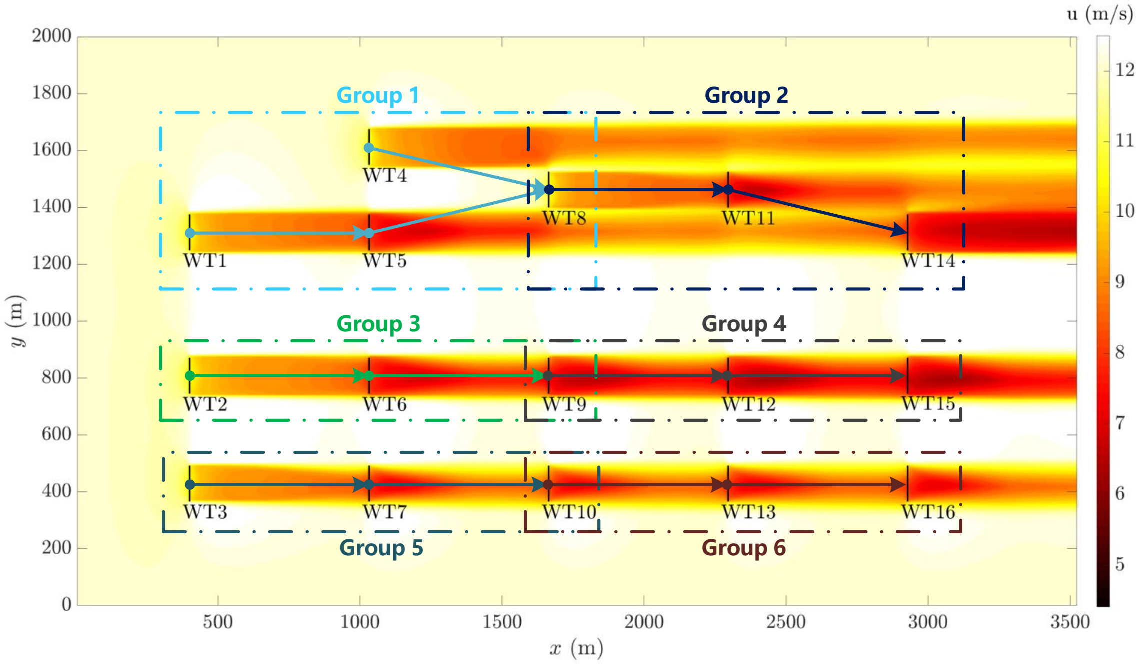

Fig. 3: Grouping Strategy Illustration.

A. Design of A Grouping Strategy for Wind Farm Control

As explained in the introduction, the aim of grouping is to divide the whole wind farm into small sub-groups based on the level of aerodynamic couplings among turbines and therefore decrease the computational complexity for DRL implementation. To this end, we introduce the following core procedures for the design of our grouping strategy.

(a) We define the turbines' influence fields to evaluate their aerodynamic couplings quantitatively. As illustrated in Fig. 1, a wind turbine will induce wakes with reduced wind speed (w.r.t to the free stream wind speed) behind it. Therefore, the deficit rate of wind speed can be employed to reflect the turbine's influence on the flow field. Based on that, we definite and formalize the influence factor of a turbine WT_i on a spatial cell $\mathcal{S}_j$ in the following equation.

$$\mathcal{IF}_{WT_i \to S_j} = \frac{1}{\beta_{\max} - \beta_{\min}} \int_{\beta_{\min}}^{\beta_{\max}} w(\beta_i) \cdot \delta U_{S_j}(\beta_i) \mathrm{d}\beta_i \tag{3}$$

where $\beta_{\max}$ and $\beta_{\min}$ denote the acceptable maximum and minimum yaw offsets of turbine WT_i, $w(\beta_i)$ is a user-defined function for weighting purposes; and $\delta U_{S_j}(\beta_i)$ denotes the deficit factor induce by the yaw offset β_i while the induction state α_i follows the greedy strategy. We calculate the integral-form factor as shown in (3) based on the turbine's yaw angle because yaw angles can significantly influence the direction of wakes and they are the main control signals for wake steering. We note that $\delta U_{S_j}(\beta_i)$ can be calculated by analytical wind farm models or directly obtained by wind farm simulators. It should be emphasized that even steady-state wind farm models can be employed to calculate $\mathcal{IF}_{WT_i \to S_j}$ because only the key features of wakes are required to achieve grouping. We employ the popular parametric model (named FLORIS) developed in [10] as an example to calculate $\mathcal{IF}_{WT_i \to S_j}$ under the setting $\beta_{\max} = 30°$ and $\beta_{\min} = -30°$. The result is given in Fig. 2, in which the warmer the color, the higher the influence factor. After that, a cut-off value of $\mathcal{IF}_{WT_i \to S_j}$ can be applied to specify the influence field.

(b) The second procedure of our grouping strategy is to build directed graphs [23] for the whole farm based on the influence field defined in (a) (we set the cut-off deficit rate to be 0.2). Specifically, every turbine in the farm is regarded as a vertex. If a downstream turbine is within the influence field of an upstream turbine, then an edge from the upstream turbine to the downstream turbine is added. Following that, the relationships of all the turbines in the farm can be described by directed graphs. We take the wind farm illustrated in Fig. 1 as an example. Based on the procedures given above, three directed graphs are deduced, as shown in Fig. 3. The turbines in each graph are summarized as follows:

Graph 1: $\{\mathcal{WT}_1 \to \mathcal{WT}_5 \to \mathcal{WT}_8 \to \mathcal{WT}_{11} \to \mathcal{WT}_{14}, \mathcal{WT}_4 \to \mathcal{WT}_8\}$

Graph 2: $\{\mathcal{WT}_2 \to \mathcal{WT}_6 \to \mathcal{WT}_9 \to \mathcal{WT}_{12} \to \mathcal{WT}_{15}\}$

Graph 3: $\{\mathcal{WT}_3 \to \mathcal{WT}_7 \to \mathcal{WT}_{10} \to \mathcal{WT}_{13} \to \mathcal{WT}_{16}\}$

(c) After building graphs for the wind farm, we are ready to get the grouping results. Actually, one can directly employ all turbines in a graph to form a group. However, this can lead to large groups due to the cascaded aerodynamic couplings among turbines, and a trade-off is required to restrain the group sizes. To this end, we divide the results from (b) into sub-graphs by restricting their depth. Counting from the root vertices (e.g., $\mathcal{WT}_1$-$\mathcal{WT}_4$ in Fig. 3), all vertices that are beyond the maximum depth are kicked out from the graph. Taking the wind farm in Fig. 3 as an example, we can get

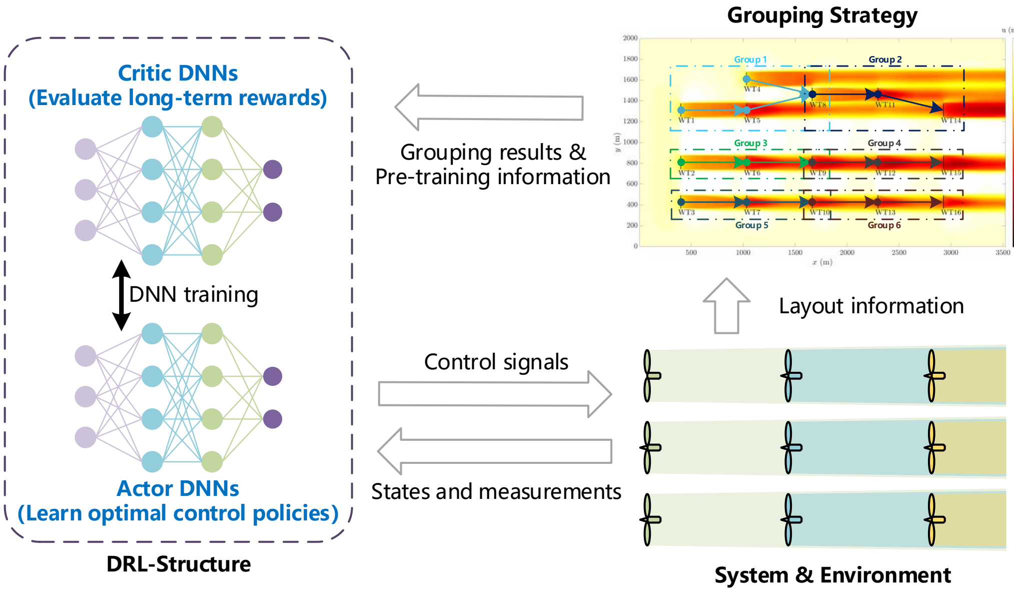

Fig. 4: Main structures of the proposed grouping-based DRL method for wind farm control.

the following sub-groups based on the cropped graphs, as shown in Fig. 3.

Sub-Group 1 & Sub-Graph 1: $\{\mathcal{WT}_1, \mathcal{WT}_4, \mathcal{WT}_5, \mathcal{WT}_8\}$

Sub-Group 3 & Sub-Graph 3: $\{\mathcal{WT}_2, \mathcal{WT}_6, \mathcal{WT}_9\}$

Sub-Group 5 & Sub-Graph 5: $\{\mathcal{WT}_3, \mathcal{WT}_7, \mathcal{WT}_{10}\}$

After that, the leaf vertices in the resulting cropped sub-graphs (e.g., $\mathcal{WT}_8$, $\mathcal{WT}_9$ and $\mathcal{WT}_{10}$ in Fig. 3) are treated as the root vertices for the remaining vertices, and the process to get additional sub-groups based on the depth restriction should be conducted over and over until the whole farm is fully grouped. For the wind farm in Fig. 3, this process leads to another three sub-groups:

Sub-Group 2 & Sub-Graph 2: $\{\mathcal{WT}_8, \mathcal{WT}_{11}, \mathcal{WT}_{14}\}$

Sub-Group 4 & Sub-Graph 4: $\{\mathcal{WT}_9, \mathcal{WT}_{12}, \mathcal{WT}_{15}\}$

Sub-Group 6 & Sub-Graph 6: $\{\mathcal{WT}_{10}, \mathcal{WT}_{13}, \mathcal{WT}_{16}\}$

It should be emphasized again that the whole grouping strategy proposed above can be adapted to wind farms with different specifications, and the one given in Fig. 3 is just a typical example to make the whole design easy to follow.

B. Design of a DRL-based Method for Wind Farm Control

Based on the grouping strategy in Sec. III.A, we design a DRL-based wind farm control method in this subsection.

Typically, reinforcement learning can be modeled by a Markovian Decision Process (MDP) [24], formalized by $\{s_k, a_k, s_k^+, r_k\}$. Here s_k denotes the system states at a time step k, a_k is the control action that transfers s_k to its successor s_k^+ at $k+1$, and r_k denotes the one-step performance metric of such a transition. The aim of RL is to maximize the long-term return defined in the following equation by providing a control policy $\mu(s)$ for any system state s_k.

$$R_k = \sum_{j=k}^{\infty} c^{j-k} r_j \tag{4}$$

where $c \in (0, 1]$ is a user-defined discount factor.

In this study, we use a typical critic-actor structure to solve the MDP problem defined above. To this end, we need to employ the famous Q-function [24], denoted by $Q_\mu(s, a)$. At a specific time point k, $Q_\mu(s_k, a_k)$ represents the long-term reward when action a_k is taken at state s_k and a control policy $\mu(s)$ is pursued thereafter [25].

The critic's aim is to learn $Q_\mu(s, a)$ by only accessible data while without requiring any system models. Such a process is based on an essential property of $Q_\mu(s, a)$:

$$Q_\mu(s_k, a_k) = r_k + cQ_\mu(s_k^+, \mu(s_k^+)) \tag{5}$$

On the other hand, the actor's objective is to find the best control policy $\mu^*(s)$ such that

$$\mu^*(s) = \arg\max_{\mu} Q_\mu(s, a) \tag{6}$$

In DRL, deep neural networks (DNN) are employed as function approximators and information processors in the critic-actor structure, and their training processes are driven by Eqs. (5) and (6) or their transformations.

Based on Eqs. (5) and (6), many popular DRL methods have been developed, such as the deep deterministic policy gradient (DDPG) method [26] and the proximal policy optimization (PPO) method [27]. Here we tweak DDPG to address the wind farm control tasks considered in this paper. The main structures of our scheme are illustrated in Fig. 4. Particularly, for any time step k, we set r_k to be the whole farm's power generation (i.e., E), and the control inputs are the change of α_i and β_i, $i = 1, 2, 3..., n$. Therefore, the control objective is to maximize the long-term farm-level power generation. It is noteworthy that all variables are normalized in our DNN training process. One can refer to [26] for the detailed implementation procedure of DDPG, particularly the ideas of experience replay and target networks.

However, it is well-known that though DDPG can achieve high-performance model-free control, it also has a relatively high computational complexity. In this study, we mitigate this issue from the following two task-oriented measures.

- Based on the grouping strategy proposed in Sec. III.A, the DRL method can be executed on the sub-groups instead of the entire farm, which can significantly reduce the overall computational cost since the control task's complexity is growing exponentially with the increase of turbine numbers.
- We employ the sub-optimal results from analytical wind farm models to carry out supervised-learning-style pre-training for the DNNs in our critic-actor structure. Then we carry out model-free fine-tuning with accessible data. Such a design not only improves the training efficiency of DRL but also allows our control scheme to take advantage of both model-based and model-free wind farm methods.

Finally, we need to address the overlap issue of different sub-groups. Based on our grouping strategy, in one specific sub-group/graph, its leaf vertices could also be included in another sub-group/graph, which may cause control policy conflicts. But an important fact is that these leaf vertices are all the most downstream turbines in the corresponding sub-groups, such as $\mathcal{WT}_8$ in the sub-group 1 for the wind farm in Fig. 3. Therefore, following a common practice in wind farm control, when applying DRL to a sub-group/graph, we can set the control policies of the most downstream turbines (i.e. leaf vertices) always to be the greedy strategy. While if these leaf vertices are also included in other sub-group/graphs, DRL-based control policies will be applied to them since they are the root vertices in these other sub-groups/graphs. This logic makes our whole design self-consistent.

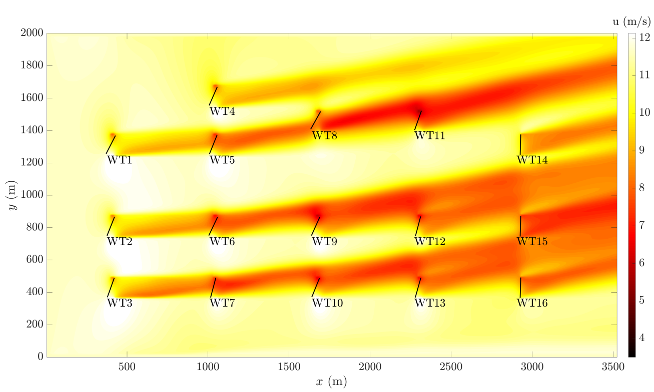

Fig. 5: Simulation results of the flow field at $t = 3000$s under GB-DRL.

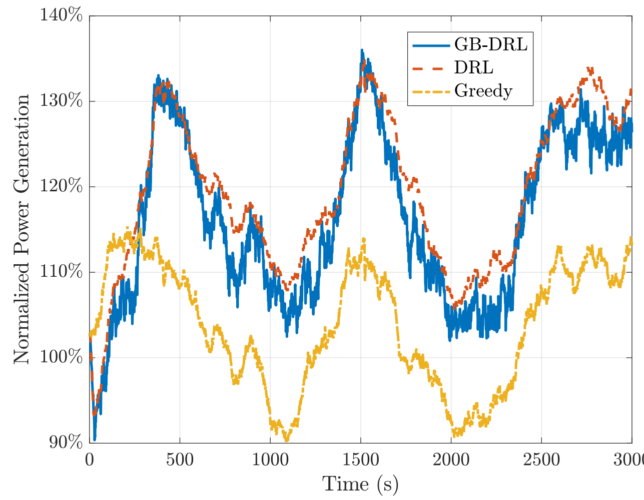

Fig. 6: Normalized power generation under different wind farm control methods.

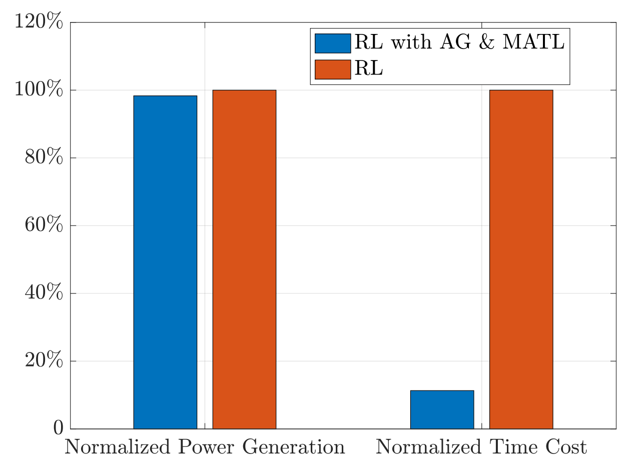

Fig. 7: Performance comparison of GB-DRL and DRL.

IV. Case Study

In this section, we employ the dynamic wind farm simulator (WFSim) developed in [22] and the 16-turbine wind farm illustrated in Fig. 3 to carry out a case study in order to show the feasibility and performance of our grouping-based DRL method (denoted by 'GB-DRL').

Based on the grouping strategy and results as given in Sec. III.A and Fig. 3, the whole farm can be divided into six sub-groups. Moreover, since the sub-groups 3-6 are homogeneous, they can be controlled by same DRL agents. In other words, we only need to apply our DRL method proposed in Sec. III.B to the sub-groups 1, 2, and 3, and then apply DRL networks same as the one for the sub-group 3 to the sub-groups 4-6. This is another advantage of grouping, which can reduce the overall computational complexity further.

We also carry out simulations for another two wind farm control methods for comparison purposes. They are

(1) The commonly-employed greedy strategy (denoted by 'Greedy'), in which every single turbine in the farm only cares about maximizing its own power generation. Greedy strategy is the benchmark in wind farm control.

(2) A DRL method (denoted by 'DRL') without the grouping strategy, i.e., it is directly executed on the whole wind

farm for power maximization. This method is employed to test the effectiveness and feasibility of our GB-DRL method.

After the training processes of DRL methods are finished, we carry out 3000-second testing simulations with time-varying free-stream wind speeds that follow a stochastic Ornstein-Uhlenbeck process. The flow field at $t = 3000$s under our GB-DRL is illustrated in Fig. 5. One can see that our method can successfully steer wakes. Simulation results of the whole farm's power generation (normalized by the power output at $t = 0$s) under all the three methods are given in Fig. 6. It can be observed that both GB-DRL and DRL can significantly increase the long-term farm-level power generation (higher by 13.09% and 16.33% on average than the greedy strategy, respectively).

We further compare the performance of GB-DRL and DRL in Fig. 7. Though applying the grouping strategy leads to a 2.54% total power generation decrease (compared with the direct implementation of DRL to the whole farm), our GB-DRL reduces 88.7% of the computing time of DRL under the same hardware & software conditions.

All these results show the feasibility and merits of our GB-DRL wind farm control method. It has the ability to increase farm-level power generation while significantly reducing the computational complexity at the cost of mild performance degradation.

V. Conclusion

This paper achieved intelligent wind farm control via deep reinforcement learning (DRL). A special grouping strategy was designed to mitigate the heavy computational complexity induced by DRL and enhance the proposed method's applicability. Based on the grouping strategy, a data-driven DRL method was proposed to be executed on the resulting subgroups instead of carrying on a complicated learning process for the entire wind farm. Simulation results indicated that the proposed method had the ability to make a trade-off between control performance and computing costs. On the one hand, it led to clearly increased farm-level power generation than the benchmark. On the other hand, it could significantly reduce the computing time at the cost of mild performance degradation compared with the direct implementation of DRL on the whole wind farm.

References

[1] G. W. E. Council, "Global Wind Report 2021," 2021.

[2] P. A. Fleming, A. Ning, P. M. Gebraad, and K. Dykes, "Wind plant system engineering through optimization of layout and yaw control," *Wind Energy*, vol. 19, no. 2, pp. 329–344, 2016.

[3] Z. Dar, K. Kar, O. Sahni, and J. H. Chow, "Windfarm power optimization using yaw angle control," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 1, pp. 104–116, 2016.

[4] M. F. Howland, S. K. Lele, and J. O. Dabiri, "Wind farm power optimization through wake steering," *Proceedings of the National Academy of Sciences*, vol. 116, no. 29, pp. 14 495–14 500, 2019.

[5] B. M. Doekemeijer, D. van der Hoek, and J.-W. van Wingerden, "Closed-loop model-based wind farm control using floris under time-varying inflow conditions," *Renewable Energy*, vol. 156, pp. 719–730, 2020.

[6] L. Dong, W. H. Lio, and F. Meng, "Wake position tracking using dynamic wake meandering model and rotor loads," *Journal of Renewable and Sustainable Energy*, vol. 13, no. 2, p. 023301, 2021.

[7] J. Zhang and X. Zhao, "Wind farm wake modeling based on deep convolutional conditional generative adversarial network," *Energy*, vol. 238, p. 121747, 2022.

[8] M. Vali, V. Petrović, S. Boersma, J.-W. van Wingerden, L. Y. Pao, and M. Kühn, "Adjoint-based model predictive control for optimal energy extraction in waked wind farms," *Control Engineering Practice*, vol. 84, pp. 48–62, 2019.

[9] J. R. Marden, S. D. Ruben, and L. Y. Pao, "A model-free approach to wind farm control using game theoretic methods," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1207–1214, 2013.

[10] P. M. O. Gebraad, F. Teeuwisse, J. Van Wingerden, P. A. Fleming, S. Ruben, J. Marden, and L. Pao, "Wind plant power optimization through yaw control using a parametric model for wake effects - a CFD simulation study," *Wind Energy*, vol. 19, no. 1, pp. 95–114, 2016.

[11] J. Park and K. H. Law, "Cooperative wind turbine control for maximizing wind farm power using sequential convex programming," *Energy Conversion and Management*, vol. 101, pp. 295–316, 2015.

[12] N. O. Jensen, "A Note on Wind Generator Interaction," 1983.

[13] J. Park and K. H. Law, "Bayesian ascent: A data-driven optimization scheme for real-time control with application to wind farm power maximization," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1655–1668, 2016.

[14] M. Vali, V. Petrović, S. Boersma, J.-W. van Wingerden, and M. Kühn, "Adjoint-based model predictive control of wind farms: Beyond the quasi steady-state power maximization," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4510–4515, 2017.

[15] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[16] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[17] H. Dong, X. Zhao, and H. Yang, "Reinforcement learning-based approximate optimal control for attitude reorientation under state constraints," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1664–1673, 2020.

[18] H. Zhao, J. Zhao, J. Qiu, G. Liang, and Z. Y. Dong, "Cooperative wind farm control with deep reinforcement learning and knowledge-assisted learning," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 6912–6921, 2020.

[19] H. Dong, J. Zhang, and X. Zhao, "Intelligent wind farm control via deep reinforcement learning and high-fidelity simulations," *Applied Energy*, vol. 292, p. 116928, 2021.

[20] H. Dong and X. Zhao, "Wind-farm power tracking via preview-based robust reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1706–1715, 2021.

[21] J. Xie, H. Dong, X. Zhao, and A. Karcanias, "Wind farm power generation control via double-network-based deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2321–2330, 2021.

[22] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, and J.-W. van Wingerden, "A control-oriented dynamic wind farm model: WFSim," *Wind Energy Science*, vol. 3, no. 1, pp. 75–95, 2018.

[23] B. Bollobás, *Modern graph theory*. Springer Science & Business Media, 2013, vol. 184.

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

[25] B. Luo, D. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only q-learning," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 10, pp. 2134–2144, 2016.

[26] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *International Conference on Machine Learning*, 2016.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.