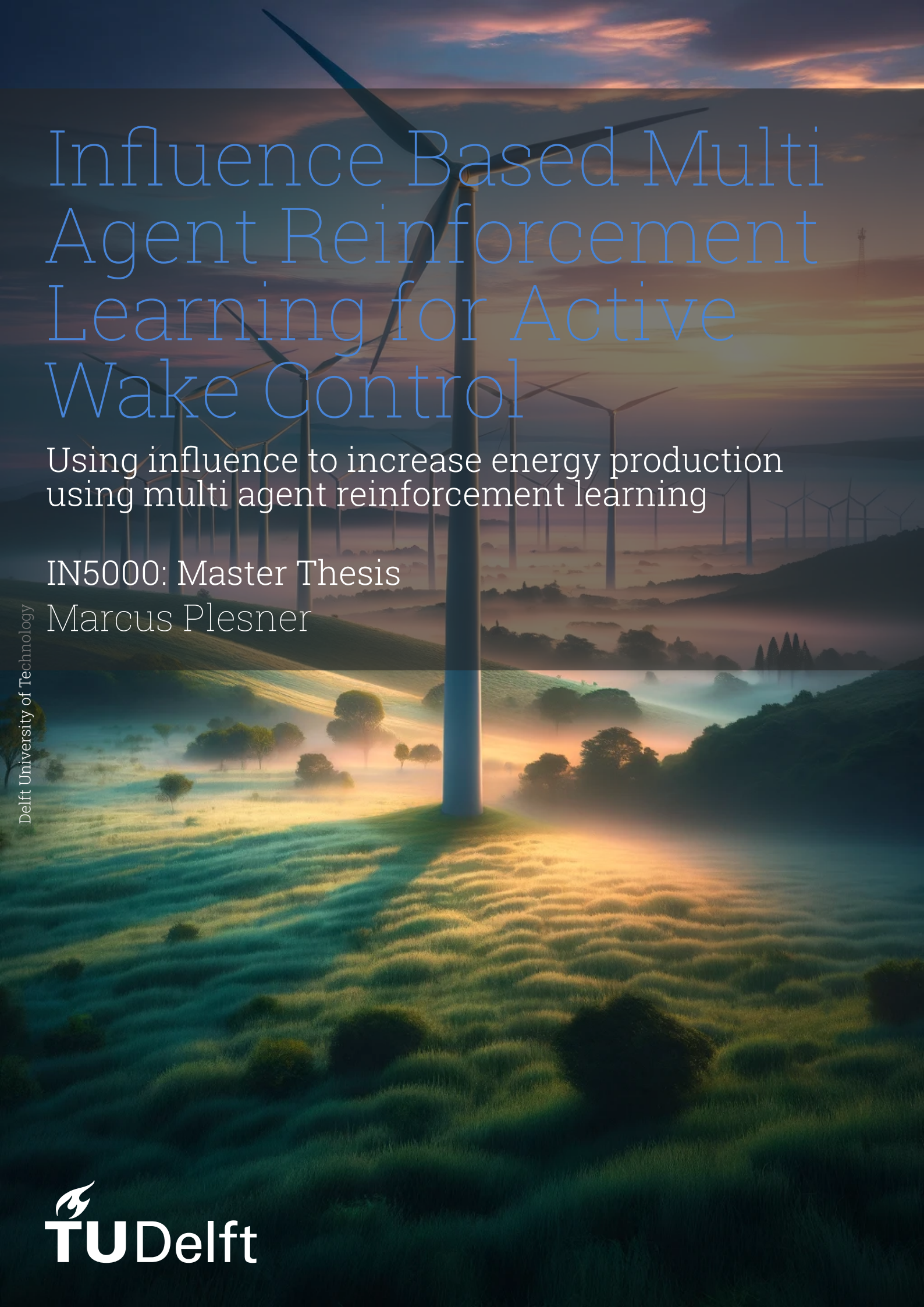

Influence Based Multi Agent Reinforcement Learning for Active Wake Control

Using influence to increase energy production using multi agent reinforcement learning

IN5000: Master Thesis

Marcus Plesner

Delft University of Technology

TU Delft

Influence Based Multi Agent Reinforcement Learning for Active Wake Control

Using influence to increase energy production using multi agent reinforcement learning

by

Marcus Plesner

Student Name	Student Number
Marcus Plesner	4932021

Supervisor: F. Oliehook
Daily Supervisor: G. Neustrov
Project Duration: November, 2023 - June, 2024
Faculty: Electrical Engineering, Mathematics, and Computer Science

Cover: DALL-E generated wind farm
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Abstract

The increasing demand for electricity has lead to demand for more efficient energy production. One promising option is wind power, which currently provides an estimated 7.8% of the world's energy production. One of the problems with wind energy is that a small percentage of the energy is lost due to the wake effect. The wake of a wind turbine is an area of low wind speed and high turbulence which is caused by the spinning of the turbine. This wake effect can mitigated by active wake control, which is a process by which the wake from a turbine is redirected away from downwind turbines, by changing the yaw of the turbine head. Calculating a policy for doing this is computationally expensive to do using numerical optimisation. Therefore, multi agent reinforcement learning is proposed to learn a policy which performs active wake control.

The proposed approach makes use of the popular reinforcement learning algorithm REINFORCE, and extends it using a variety of methods. First, a simplified version of the problem is treated, wherein the wind direction is fixed. Then the problem is made more realistic by introducing changing wind directions. The first extension of REINFORCE that is treated is difference rewards, a reward shaping strategy which seeks to solve the credit assignment problem, thereby improving cooperation between turbines. The second method uses training regimes, which train different agents at different times to stabilise the environment as much as possible. Next, role-based reinforcement learning is used to conteract the complexity of the problem by allowing each agent to specialise for a certain role. Finally, since roles cannot be manually determined for larger farms, influence-based abstraction is used to enable agents to learn the roles themselves, by abstracting spacial information and presenting it to the agent as an observation.

The results demonstrate that multi agent reinforcement learning can be used to perform active wake control in wind farms. Furthermore, the extensions proposed are shown to improve learning, and lead to greater energy output. While multi agent reinforcement learning is shown to be a promising way to tackle active wake control in wind farms, research is needed to improve the stability of the learned policies.

Contents

List of Figures

Nomenclature

Abbreviations

Abbreviation	Definition
AWC	Active Wake Control
FLORIS	FLOw Redirection and Induction in Steady State
RL	Reinforcement Learning
MARL	Multi Agent Reinforcement Leraning
OU	Ornstein-Uhlenbeck
IBA	Influence-Based Abstraction

1 Introduction

Increasing the energy yield from renewable sources contributes to the reduction of reliance on fossil fuels, which lead to greenhouse gas emissions. Given the environmental and economic incentives, the implementation of more efficient green technologies presents a dual benefit: it enhances the efficiency and output of renewable energy sources while concurrently decreasing the environmental footprint associated with energy production. In certain environments, such as the United States midwest, and northern Europe, wind energy is a promising green alternative to fossil fuels [1].

1.1. Active Wake Control

The wake from a wind turbine is the region of disturbed airflow behind the turbine, characterized by reduced wind speed and increased turbulence. As the turbine blades spin from the wind to generate electricity, they create a zone of lower velocity and turbulent flow patterns. This wake effect can impact downstream turbines in a wind farm, leading to reduced efficiency and increased mechanical stress. Understanding and mitigating wake effects is crucial for optimizing the performance of wind farms.

The necessity for sustainable energy solutions has led to the exploration of innovative techniques to enhance the efficiency of wind farms, amongst which active wake control (AWC) emerges as a possible advancement. As wind turbines spin, they generate areas behind them with slow wind speed and high turbulence, called the wake. The wake lowers the power output of downwind turbines, thereby diminishing the farm wide energy output by 7-13% [17, 16, 7]. AWC addresses this issue by adjusting the yaw angle of upstream turbines, directing the wakes away from the path of downwind turbines and mitigating wake-induced losses [29]. The process is a cooperative effort, in which the upwind turbines sacrifice their own energy output to increase the output of downwind turbines.

Wind energy is a key component in the modern and rapidly expanding domain of electricity production from renewable sources. The drive behind the ongoing development of wind power is due to its renewable nature, positioning it as a solution to global energy demand [38]. Characterized by its minimal pollution, the wind energy sector is actively addressing ancillary issues such as noise pollution and wildlife collisions, further solidifying its role in reducing dependence on the global oil market. The industry not only fosters job creation, especially in rural locales where wind farms are predominantly established, which can improve local economies. Despite these advantages, wind energy contributes a mere 7.8% [18] to global electricity demand, a limitation attributed to its relatively low power production density when compared to other energy sources. This shortfall is primarily due to the modest rated power of even the largest horizontal axis wind turbines, and the difficulty of installation. Furthermore, 7-13% of energy is lost due to the wake effect [7, 16]. The wake effect is relatively minimal for small wind farms, which often consist of a single row of turbines which is placed with respect to the primary wind direction at the site. Wake effects are significantly more important for larger wind farms, since they are often arranged in a grid layout, which results in frequent wake effects between turbines.

Addressing the critical issue of wake effects is an important area research within the wind energy sector, since is a way to increase the potential of future wind farms, but also because it can improve the

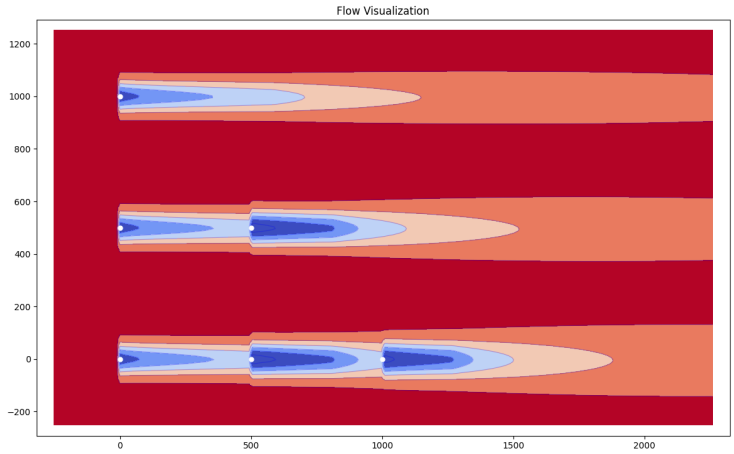

Figure 1.1: Wake Effect

performance of already existing wind farms at no cost. Strategies to mitigate wake effects have varied from modifying blade rotation directions to optimizing wind farm layouts. Active control techniques, aimed at manipulating the wake, have emerged as effective solutions to enhance the annual energy production of existing wind farms. Techniques such as controlling blade pitch and generator torque to weaken the wake, inducing faster wake recovery, and redirecting wakes away from downstream turbines through yaw misalignment, demonstrate promising results. The most well studied method of active wake control is through yaw manipulation, in which wakes are redirected away from downstream turbines by changing the yaw of the upwind turbine. The problem is developing a policy which dictates when to change the yaw, in which direction, and by how much.

Numerical optimisers combined with simulation tools have previously been used to optimise energy output. In research, steady state simulators such as FLORIS (Flow Redirection and Induction in Steady-State), have been used. Large eddy simulation is a more advanced form of simulation which uses fluid dynamics to simulate the propagation of wakes, which allows for a more realistic analysis of wake interactions. The higher fidelity simulation comes at the cost of higher computational complexity. Therefore this technique is normally used for smaller farms [46], and even in smaller farms it can require significant computational resources [4]. Large eddy simulation has also been used for larger offshore wind farms [5], but this requires significant computational resources. To mitigate some of the computational complexity and introduce some flexibility, reinforcement learning (RL) has been proposed for AWC.

The challenge of performing AWC in wind farms is caused by the computational intensity required to model wakes and energy production accurately. The complexity leads to difficulty constructing a policy that controls the turbines such that energy production is maximised. [50]. Wakes are inherently nonlinear phenomena, rendering their simulation a complex and resource-intensive task. For instance, optimizing the wake control strategy for a modestly sized farm with 14 turbines using numerical methods and large eddy simulations can take as long as five days [4]. This computational demand scales significantly for modern wind farms, the biggest of which consisting of over 150 turbines, such as the Hornsea projects, underscoring the need for more efficient optimization techniques. Moreover, the variable and unpredictable nature of wind introduces further complexity, leading to the need for policies capable of adapting to complex systems.

One promising approach which can address these challenges is RL, with its ability to generate adaptable policies through the treatment of the problem as a Markov decision process, is potentially a suitable method for addressing the dynamic challenges presented by wind farm optimization.

1.2. Reinforcement Learning

The exploration of RL for reducing wake-induced losses in wind farms is a recent development, building upon traditional approaches like layout optimization and numerical optimization [45, 30]. Previous studies have demonstrated the potential of RL in optimizing the performance of smaller wind farms consisting of 10 to 15 turbines [31].

RL is a type of machine learning where an agent interacts with an environment to in order to maximise a given reward function [37, 43]. The agent takes actions and receives feedback in the form of rewards or penalties. Using this feedback, it learns the best policy, to accumulate the highest possible reward over time. This is often formalised as a Markov decision process, where the an action is taken based on the current state of the environment [8, 51]. This process of exploration and feedback, combined with long-term reward maximization, enables the agent to develop a policy for solving complex problems. RL is distinguished by its focus on learning from the consequences of actions, rather than from direct instruction, making it applicable to a wide range of tasks from robotics to game playing.

RL may be effective for AWC because it has been shown to be successfull in various continuous control tasks, such as autonomous driving [22]. It also enables accurate modelling of a wind farm because turbines can be modelled as agents, which can be trained individually or as a group. RL has been used in smaller wind farms, with a wind tunnel setup, which is where the wind direction is fixed for the duration for the duration of the experiment. This has been successful, but scalability and changing wind directions remain an issue. While promising, RL is limited by the instability of the algorithms, who's performance may be unreliable, or start to deterioriate. This makes current RL algorithms unsuitable for real life wind farm control without modification.

RL was traditionally limited by small state spaces, but it has evolved significantly with the advent of function approximation techniques using deep learning. Deep Q-networks, as introduced by Mnih et al. [28], enable the estimation of state-action, so called, Q-values for various actions across generalized states, showcasing success in complex tasks such as playing Atari games without prior knowledge. This method has set new standards in RL, with numerous extensions further broadening its applicability. In contrast to deep Q-networks, that focus on learning the value of actions, policy gradient methods [52], directly adjust the policy based on the expected rewards. This offers a more direct path for actions selection in environments with continuous or complex action spaces.

To extend the use cases of RL, multi-agent reinforcement learning (MARL) was proposed [2]. MARL adapts the principles of RL to scenarios involving multiple agents, making it possible for agents to work together in a cooperative way [25]. This approach facilitates a more flexible optimization strategy, allowing for the adjustment of individual agent behaviors to collectively enhance the policy. MARL is applicable to wind farms since each turbine can be modelled as an individual agent which needs to optimize its performance while considering its influence on other turbines. Through the application of MARL algorithms, wind farms can potentially achieve significant improvements in energy production.

Influence in MARL pertains to the dynamics where the actions of one agent affect the learning and behavior of other agents within the same environment. This concept is useful in systems where multiple agents interact and learn concurrently. The study of influence in MARL seeks to understand and model how agents can optimally adapt their strategies not only based on the static properties of the environment but also in response to the evolving strategies of other agents. This inter-agent influence can significantly complicate the learning process, as each agent must consider not only the immediate rewards of its actions but also the potential reactions and adaptations of others. Consequently, influence modeling in MARL involves developing algorithms that can effectively handle these interactions, ensuring that agents can learn cooperative or competitive strategies that are robust and scalable. Understanding influence in this context is useful for designing systems that can operate efficiently in complex, dynamic, and often uncertain environments.

1.3. Problem Statement

The integration of MARL with AWC, presents a novel approach to increasing energy production of wind farms. In this context, each wind turbine functions as an autonomous agent that interacts with the complex environment of the wind farm. The primary objective is to enable these turbines, through MARL, to learn a policy which maximises the long term energy production by mitigating wake induced

losses. Such adjustments aim to mitigate wake effects—turbulent airflows generated by upstream turbines that can significantly reduce the energy production of downstream turbines. By leveraging MARL, turbines can collectively optimize wake orientation, thus maximizing the overall energy output of the farm. Given that the primary difficulty is in learning the influence relationships between turbines, and acting accordingly, the concept of influence in RL will be used to improve learning.

1.4. Research Question

The aim of this thesis is to improve upon MARL algorithms using influence information and domain knowledge for AWC in wind farms, enabling turbines to autonomously optimize their operational policies to maximise overall energy production. This framework seeks to investigate the potential of applying MARL to address the complex dynamics of wake effects in wind farms, offering novel approaches to increase efficiency and energy output. To explore the viability of MARL for advanced control systems in renewable energy generation, we propose the following central research question:

Main Research Question

How can influence information and domain knowledge be used with multi agent reinforcement learning algorithms to increase wind farm energy production through active wake control

From this research question we can derive several subquestions regarding how this is accomplished, using the concept of influence.

Subquestion 1

How can domain knowledge of wake propagation be used to design reward functions which encourage cooperation between agents?

The design of the reward function is crucial in shaping the behaviors of agents in MARL applications, especially in scenarios like AWC for wind farms, where the goal is to increase collective energy production. This is a version of the credit assignment problem [26], which is the problem of attributing rewards in proportion to each agent's contribution to performance. The challenge lies in formulating a reward mechanism that promotes not just individual optimization by each turbine but leads to a level of cooperation essential for mitigating wake effects across the farm efficiently. To this end, the energy output of the turbines which are influenced should count towards an upwind turbine's reward function.

A new approach to defining rewards could involve not only the immediate energy production gains of individual turbines but also account for the downstream effect of their actions on the farm's total energy yield. By quantifying the reduction in wake losses and increased overall efficiency as part of the reward, turbines can be incentivized to adopt strategies that may sacrifice their maximal instantaneous output for greater collective gains. Such a reward structure could be dynamically calculated by running multiple simulations, taking into account real-time conditions, thereby making the consequences of cooperative versus greedy actions explicitly clear to each agent. By emphasizing the global optimization of energy production, this methodology can accelerate the convergence speed of the MARL algorithms, leading to more effective and efficient wake management strategies within wind farms.

Subquestion 2

How can training regimes be used to stabilise and improve performance?

Training regimes can play a role in enhancing and stabilizing the performance of MARL systems, in applications like AWC, where the interaction dynamics between agents are complex. Effective training regimes are designed to stabilise the environment, enabling the agent to optimise it's policy in the simplest possible learning scenario. This approach ensures that agents not only learn optimal strategies for a broad spectrum of conditions but also develop the resilience needed to adapt to unforeseen changes and disturbances in the environment. The primary aim is to allow the agent to optimise its policy in a stable environment, where the other agents' actions are factored out. This avoids the moving targets problem, and allows each agent to learn its effects on the whole farm.

Training agents in more stationary environments, where the dynamics remain consistent over time, mitigates the moving targets problem inherent in many multi-agent settings. In such stabilised

environments, the actions of other agents change less unpredictably, allowing an individual agent to learn and adapt to the changing environmental factors. This stability enables agents to discern the impact of their actions on the overall performance more clearly, fostering a deeper understanding of their role within the collective system. As a result, agents can refine their strategies more effectively, optimizing their contributions towards achieving shared objectives and enhancing collective performance through coordinated efforts.

Subquestion 3

Can domain knowledge be used to assign task specific roles to agents to speed up and stabilize learning in active wake control?

Role-based reinforcement learning is a framework within multi-agent systems where each agent is assigned a specific role or task within the system. This specialized approach simplifies the learning process by reducing the complexity that each agent needs to handle. Instead of every agent needing to learn the full scope of possible behaviors and strategies in a dynamic environment, each one focuses on mastering the specific tasks and responses associated with its role. This can accelerate the learning process by limiting the range of actions and scenarios each agent needs to consider but also enhances the efficiency of the overall system. Having each agent focus on a single task or a subset of tasks minimizes overlap in responsibilities, reduces conflicts, and ensures a more coordinated and coherent behavior across the system.

In the context of wind farms, the role of a turbine is primarily determined by its position relative to others, the direction of the wind, and consequently, how many other turbines it impacts with its wake. Turbines that are positioned upstream, for instance, play a the central role in wake steering to optimize airflow for downstream turbines, whereas downwind turbines always have to face the wind. By assigning roles based on the number of turbines which a given turbine influences, the learning task for each agent is simplified Each agent can optimize its strategies based on a single task, and more learning samples, which should decrease the complexity of the task. This specialised approach allows agents more opportunity to learn the correct policy.

Subquestion 4

How can influence-based state abstraction be used to learn wind turbine roles in wind farms?

Influence-based state abstraction represents a way of managing complexity in optimisation tasks. State abstractions do this by communicating underlying information in the state explicitly, thereby simplifying learning. This is done by creating a simplification of information presented in the agents' observations, which still captures the necessary information. Since it is often impossible to only capture the relevant information in an abstraction, the quality of the state abstraction is based on how much of the relevant information it captures, and how much irrelevant information it removes.

In the context of wind farms, different environment states require the same policy, and explicitly providing this information in the observations can help the agents learn their roles, based on the environmental conditions. This is especially important for larger wind farms, in which roles cannot be manually determined. This can be useful for larger wind farms, where the number of roles may be too large to discern. By abstracting the influence of individual turbines on the farm's collective output, the agent can identify patterns of symmetry that allow for the scaling of efficient solutions across environmental conditions. This approach aims to enhance energy production and stabilise the learning process by providing the agent with the most important information.

1.5. Contributions

This research makes contributions to the field of AWC and RL through the exploration and application of advanced reinforcement learning techniques. An achievement is the exploration of policy gradient methods for AWC, which have shown themselves to be suitable for complex multi agent environments. By adapting policy gradient approaches, which directly optimize the policy function rather than the value function, this work demonstrates how agents can learn to navigate the complex dynamics of wakes in wind farms more effectively. This exploration not only advances the theoretical framework for applying reinforcement learning in renewable energy contexts but also sets a precedent for future

investigations into environmentally sustainable energy optimizations.

Furthermore, this research has tested and developed reward differencing methods to enhance performance in AWC tasks. By designing these methods to identify the impact of individual turbines on the overall efficiency of the farm, the study successfully addresses the challenge of credit assignment in multi-agent environments. Additionally, the introduction of parallelization in the differencing methods is a step towards reducing computational demands, significantly decreasing computation time without compromising the integrity of the learning process. Coupled with the development and testing of novel training regimes aimed at improving learning stability, this research presents a comprehensive suite of advancements. These training regimes, mitigate the volatility often associated with reinforcement learning in complex, dynamic settings, ensure more robust and reliable agent performance. Finally, role-based reinforcement learning simplifies the learning process by ensuring that an agent has one job to learn and thereby learns the simplest possible policy. The use of influence-based state abstraction helps agents learn its role from simplified information, while providing explicit information about environmental symmetries. Together, these contributions lead so the successful application of reinforcement learning in AWC and lay a foundation for future research in the efficient and intelligent management of wind energy resources.

1.6. Outline

This thesis develops by first laying the groundwork in Chapter 2: Preliminaries, where we discus the essentials of RL and its applications in renewable energy. Chapter 3: Experimental Setup then describes the simulation environment, detailing the setup for modeling wind farms and turbine interactions, using the FLORIS framework. In Chapter 4: Difference Rewards, we introduce and implement a novel reward system to promote cooperative behaviors among turbines, followed by an evaluation of its effectiveness. Chapter 5: Staged Training discusses the adoption of training strategies to enhance the MARL models' learning process and stability. Chapter 6: Role-Based Reinforcement Learning shows how the use of domain knowledge to train task specific agents can simplify the complexity of the environment. Chapter 7: Influence-Based Abstraction introduces state abstractions to identify symmetries accross environment states. Finally, Chapter 8: Conclusion and Further Work wraps up the study with a summary of findings, reflections on the challenges, and suggestions for future research avenues, aiming to advance the efficiency of wind farms through improved AWC techniques.

2

Preliminaries

In this chapter, the prerequisite information about the technologies used will be given. First, FLORIS is discussed as a tool for modeling wake interactions within wind farms. FLORIS provides a simple and computationally efficient framework for simulating turbine wake effects and their impact on energy production. Next, we will formally examine MARL, as a framework solving optimisation problems with multiple agents MARL is particularly relevant to scenarios involving distributed systems, such as wind farms, where coordinated decision-making is essential. Finally, the REINFORCE algorithm, which is a fundamental policy gradient method in RL is treated.

2.1. FLORIS

FLORIS is a steady-state wake modeling framework that integrates several state-of-the-art steady-state models. It provides tools for the analysis and optimization of wind farm layout and operation, developed to be open source, computationally inexpensive, and implemented in Python. According to Fleming et al. [14] FLORIS is robust, enabling precise control and optimization strategies in wind farm management by modeling wind flow and turbine interactions efficiently.

The effectiveness and precision of FLORIS have been confirmed through various research studies and practical applications. Gebraad et al. [15] have applied a FLORIS-based control strategy in a high-fidelity computational fluid dynamics simulator, demonstrating its capability to adapt to complex simulation environments. Furthermore, Schreiber et al. [39] and Fleming et al. [14] have conducted wind tunnel tests and field trials in commercial offshore wind farms to validate FLORIS's accuracy in real-world settings. These studies collectively highlight FLORIS's role as a leading tool in both wake modeling and model-based active wake control, making it a preferred choice among researchers.

An illustrative example of FLORIS's application in operational optimization was provided through a field test at an offshore wind farm [14]. In this study, the normal yaw controller of an array of turbines within an operating commercial offshore wind farm in China was modified to implement wake steering according to a yaw control strategy. This strategy, designed using the computational models from the National Renewable Energy Laboratory, including FLORIS, aimed to deflect turbine wakes in a manner that would increase the overall power output of the wind farm. The results from this experiment were promising. The wake-steering controller successfully increased power capture, with outcomes closely aligning with predictions derived from FLORIS and other simulation models. This success not only demonstrates FLORIS's practical applicability in real-world settings but also reinforces the model's reliability and effectiveness in optimizing wind farm operations.

Moreover, the integration of FLORIS in wind farm management practices highlights its capacity to support the development of advanced control strategies. These strategies leverage detailed wake dynamics provided by FLORIS to enhance the efficiency of wind farms under varying atmospheric conditions. By enabling precise manipulation of turbine wakes, FLORIS helps in mitigating wake-induced power losses and improving the overall energy output from wind farms.

FLORIS's utility extends beyond mere theoretical modeling. FLORIS offers significant capabilities in simulating various atmospheric conditions, making it an ideal tool for RL applications in wind farm management. By accurately modeling different wind speeds, directions, turbulence intensities, and other atmospheric variables, FLORIS provides a realistic and dynamic environment in which RL algorithms can be trained and tested. The ability to replicate diverse weather scenarios allows RL models to learn and optimize turbine control strategies under a wide range of conditions, enhancing their adaptability and effectiveness in real-world settings. RL can leverage FLORIS-generated data to train control algorithms that dynamically adjust turbine yaw settings to minimize wake interference and maximize power output. Moreover, the computational efficiency of FLORIS enables the frequent iteration and updating of RL models, facilitating rapid learning cycles that are essential for refining control strategies in response to fluctuating environmental conditions. The integration of FLORIS with RL techniques is a natural first step to developing RL based control systems which continuously enhance their performance by learning from a large dataset of simulated atmospheric conditions.

2.2. Multi Agent Reinforcement Learning

MARL extends the framework of RL to environments where multiple agents interact simultaneously, each seeking to optimize the group's performance through trial and error [2]. In MARL, agents learn to make decisions by observing the state of the environment, executing actions, and receiving feedback in the form of rewards. These interactions are modeled as a Markov Decision Process for each agent, where the transition from one state to another is dependent only on the current state and the action taken, not on the sequence of events that preceded it. One of the main advantages of MARL is its ability to tackle complex problems that involve multiple decision-makers, such as traffic light control, collaborative robotics, and economic simulations [40, 47, 44]. However, the complexity in MARL arises because the environment's state and the rewards an agent receives can be affected by the actions of other agents [33]. This interdependence requires agents to adapt their strategies not only to the dynamic aspects of the environment but also to the dynamic behaviors of other agents, making the learning process significantly more complex.

Diverging from traditional single-agent RL, where an agent's action influences the environment in a relatively predictable manner, MARL involves considerations of other agents' actions, leading to a more complex optimisation problem. This inherent complexity introduces unique challenges, such as the need for agents to cooperate effectively within the same environment. One of the main problems in MARL is assigning reward or penalty in correct proportion to the individual agent's performance. This is a problem because agents can act in ways which confound the reward signal, leading to lower performance [33] Furthermore, the environment can be modified by other agents which negatively effect performance if the problem is not mitigated through communication between agents.

The multi agent setting can be modelled as a multi agent Markov decision process [8, 24], where $M = \langle D, S, \{A^i\}_{i=1}^{|D|}, T, R, \gamma \rangle$ where $D = \{1, ..., N\}$ is the set of agents; $s \in S$ is the state; $a^i \in A^i$ is the action taken by agent i and $a = \langle a^1, ..., a^N \rangle \in \times_{i=1}^{|D|} A^i = A$ denotes the joint action; $T(s'|a, s) : S \times A \times S \rightarrow [0, 1]$ is the transition function that determines the probability of ending up in state s' from s under joint action a; $R(s, a) : S \times A \rightarrow R$ is the shared reward function and γ is the discount factor.

Agent i selects actions using a stochastic policy $\pi_{\theta^i}(a^i|s) : S \times A^i \rightarrow [0, 1]$ with parameters θ^i, with $\theta = \langle \theta^1, ..., \theta^N \rangle$ and $\pi_\theta = \langle \pi_{\theta^1}, ..., \pi_{\theta^N} \rangle$ denoting the joint parameters and policy, respectively. With r_t is the reward at time t and expectations taken over episodes. The policy π_θ has the value function:

$$V^{\pi_\theta}(s_t) = \mathbb{E}_{\pi_\theta}[\sum_{l=0}^{\infty} \gamma^l r_{t+l} | s_t]$$

and action-value function:

$$Q^{\pi_\theta}(s_t, a_t) = \mathbb{E}_{\pi_\theta}[\sum_{l=0}^{\infty} \gamma^l r_{t+l} | s_t].$$

At each time step, the agents try to maximize the value function $V^{\pi_\theta}(s_t)$.

2.3. REINFORCE

Policy gradient methods represent a class of algorithms within the realm of RL, enabling agents to directly learn an optimal policy that dictates the probability distribution of actions in given states to maximize long-term rewards [52]. Unlike value-based approaches, which focus on estimating the value of actions to derive a policy, policy gradient methods optimize the policy directly through gradient ascent. This is achieved by calculating the gradient of the expected reward with respect to the policy parameters and adjusting the parameters in the direction of the gradient to increase the probability of beneficial actions. One of the fundamental algorithms in this category is REINFORCE, which operates by sampling actions according to the current policy and adjusting the policy parameters via gradient ascent using the returns as an indicator of the direction and magnitude of the change. This method allows for the learning of policies in environments with both discrete and continuous action spaces, expanding the applicability of reinforcement learning to a broader range of complex problems.

Further enhancing the efficiency of policy gradient methods, REINFORCE with baseline introduces a modification aimed at reducing the variance of the gradient estimate, thus accelerating the learning process. The baseline, which is a function of the current state, does not affect the expected value of the gradient but reduces its variance by normalizing the rewards. This technique is particularly useful in scenarios where the scale of rewards is large or highly variable, as it helps in stabilizing the training. By subtracting a baseline value from the calculated returns before updating the policy parameters, the algorithm can more effectively discriminate between better and worse actions, leading to more stable and faster convergence. The combination of policy gradient methods, particularly REINFORCE and its enhancement with a baseline, is a robust framework for solving decision-making tasks that are too complex for traditional value-based methods.

REINFORCE is a policy gradient method, which maximises the expected value function [52], by directly optimizing the policy parameters θ. It achieves this by performing gradient ascent in the direction that maximizes the following expected value function.

$$V(\theta) = \mathbb{E}_{s_0}[V^{\pi_\theta}(s_0)]$$

REINFORCE is the simplest policy gradient method, which executes the current policy π_θ for an episode of T steps and optimises it with the following update:

$$\theta \leftarrow \theta + \alpha \sum_{t=0}^{T-1} \gamma^t G_t \nabla_\theta \log \pi_\theta(a_t|s_t)$$

where

$$G_t = \sum_{l=0}^{T-t-1} \gamma^l r_{t+l}$$

is an unbiased esitmate of $V^{\pi_\theta}(s_t)$ computed over the episode. This update rule corresponds to performing stochastic gradient ascent on $V(\theta)$ because the expectation of the update target is the gradient of the value function:

$$\mathbb{E}_{\pi_\theta}[\hat{G}] = \nabla_\theta V(\theta)$$

Given an appropriate choice of step size α, the method will converge. The following pseudocode shows the process, which is repeated at each episode. First the episode is run, then the log-probabilities of the eactions is calculated. Finally the update step previously described is performed

REINFORCE suffers from the high variance of the sampled returns due to the stochasticity of the environment and the policy iteself. Therefore it converges slowly. To remedy this, a suitable baseline $b(s)$, can be subtracted from the return G_t to reduce the variance [43]. This baseline provides an estimate of the expected return for a given state, which helps in assessing whether the action taken in that state was better or worse than expected. By subtracting this baseline from the return, REINFORCE can focus on learning the relative advantage of each action rather than just the raw return itself. This technique, known as baseline subtraction, can significantly stabilize the learning process and accelerate convergence, making REINFORCE more efficient in various reinforcement learning tasks.

Algorithm 1 REINFORCE

Initialize policy π_θ
for episode in 1..NumEpisodes **do**
  collect a trajectory by following π_θ
  calculate log-probability of action at each time step
  calculate Return at each time step
  calculate $\nabla_\theta J(\theta) = \nabla_\theta \sum_{t=0}^{T} \pi_\theta(a_t|s_t) R_t$
  $\theta = \theta + \alpha \nabla_\theta J(\theta)$
end for

3

Experimental Setup

In this chapter, we delve into the details of the experimental setup, detailing the methodologies and tools employed to explore active wake control. We begin by examining different configurations of wind farm layouts utilized in the study, considering how they can lead to a better understanding of wake effects. Following this, we explore the various wind processes will be used in the study, as well as how they are generated, and the reasons for using them. Central to our investigation is the use of the FLORIS wake modelling library, which enables us to create a reinforcement learning environment. This simulator is necessary for modeling the dynamic interactions between turbines, thereby facilitating the development and testing of reinforcement learning algorithms designed to optimise wind farm performance.

3.1. Layouts

The first layout which is used in this research is a triangle arrangement with turbines configured in descending rows of three, two, and one from the most upwind position, offers a unique approach to studying and managing wake interactions across multiple depths. This layout's tiered structure facilitates an analysis of how wake effects propagate and differ depending on the spacing and alignment of the turbines. The advantage of having turbines arranged in a triangle formation is the ability to observe the cumulative impacts of wake as it moves through rows of varying depth. This setup simulates real-world scenarios where turbines are frequently layed out on a grid, which can lead to a large number of different row depths. Moreover, this layout allows for the assessment of turbine performance variability due to wake effects at varying distances, which is critical for optimizing the placement and operation of turbines in a way that maximizes the overall efficiency and power output of the wind farm.

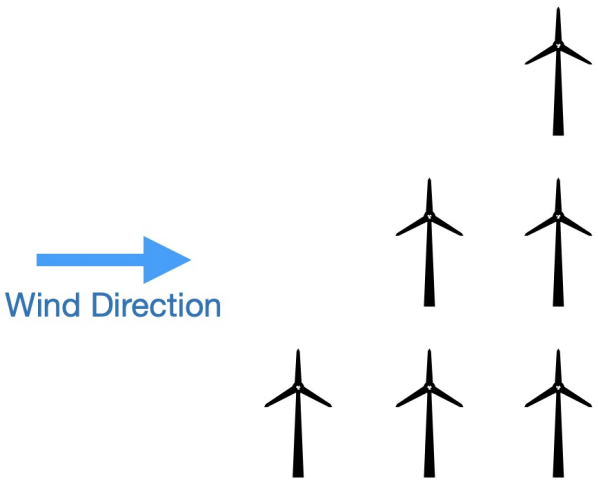

Figure 3.1: Triangle wind park layout

The second layout features a hexagonal configuration with a central turbine surrounded by others on all sides. This design is particularly beneficial for studying the dynamics of wake interaction in a densely packed environment where the central turbine is likely to affect the performance of surrounding turbines from multiple angles. The central placement ensures that no matter the wind direction, wake interactions occur, providing a richer dataset for analyzing the impact of wakes under various wind conditions. This layout is advantageous for developing control strategies that enhance the resilience and efficiency of wind farms by dynamically adjusting turbine operations to mitigate detrimental wake effects. The hexagon layout mimics wind farm designs and is useful for testing advanced wake steering techniques that can be applied to real-world settings where multi-directional wake effects are prominent.

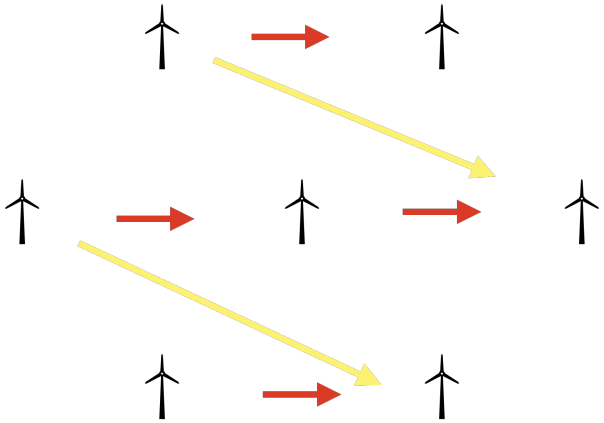

Figure 3.2: Hexagon wind park layout

Figure 3.2 shows the wind directions which will cause wake induced losses. The red arrows show a direction which would cause the most loss if the turbines all faced the wind, since the largest number of turbines is affected and the closest distance. The yellow arrows show another less detrimental scenario where fewer turbines are influenced and at a greater distance. The directions shown are symmetrical across each 60° arc.

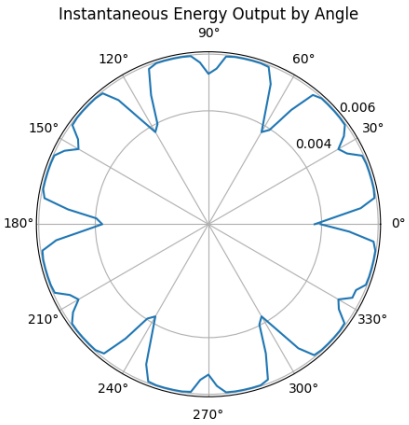

Figure 3.3: Energy production of hexagonal wind farm by wind direction angle

The hexagonal layout has symetrical energy losses for different angles as shown in figure 3.3. The plot shows the instantaneous energy production at each angle. As shown, the deeper troughs,in which energy production decreases by approximately 30%, correspond to the red arrows in figure 3.2. The yellow arrows in figure 3.2 correspond to the smaller troughs in figure 3.3.

3.2. Wind Processes

In the first wind process scenario, the constant wind direction, coupled with wind speed and turbulence generated by Ornstein-Uhlenbeck (OU) processes, provides a controlled environment ideal for fixed

direction wind experiments. In future chapters, this process will be refered to as the fixed-direction process. This setup is widely utilized to test the efficiency of different turbine alignments and operational strategies under steady directional flow conditions. The primary benefit of maintaining a constant wind direction is the ability to isolate the effects of varying wind speeds and turbulence on turbine performance, thus enabling a clearer analysis of how these factors influence energy output and wake behavior. By employing the OU process to simulate natural fluctuations in wind speed and turbulence, real-world wind patterns are replicated in a controlled setting. This approach is necessary to determine whether adaptive control strategies can outperform a naive 'face-the-wind' approach, where turbines are simply aligned with the prevailing wind without adjustment. Moreover, this method tests the ability of control algorithms to converge towards an optimal strategy for maximizing energy capture and minimizing wake-induced losses for a given wind direction. Such studies are the first step in advancing our understanding of optimal turbine policy, that can adjust in real-time to changes in wind conditions.

The OU process is a stochastic process used to model mean-reverting behavior in continuous time. It is characterized by its drift and diffusion components. The drift term, typically proportional to the difference between the current value and a long-term mean, drives the process back towards this mean, reflecting the mean-reverting property. The diffusion term, representing random fluctuations, introduces randomness into the process. Mathematically, it is defined by the stochastic differential equation below where θ is the rate of reversion, μ is the long-term mean, σ is the volatility, and W_t is a Wiener process.

$$dx_t = -\theta(\mu - x_t)dt + \sigma dW_t$$

The OU process is highly versatile for modeling different types of mean-reverting behaviors by adjusting its parameters. For instance, as demonstrated in the provided plots, the left graph depicts an OU process with a low mean and high mean reversion rate, resulting in values that quickly revert to the mean and exhibit less variability around it. In contrast, the right graph shows an OU process with a higher mean, increased volatility, and lower mean reversion rate, leading to broader fluctuations and a slower return to the mean. This flexibility in parameter adjustment makes the OU process suitable for a wide range of applications, allowing for tailored modeling based on specific requirements.

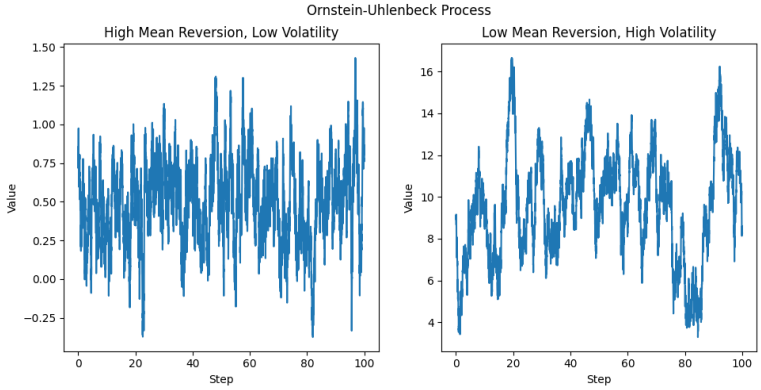

Figure 3.4: Ornstein-Uhlenbeck process examples

The second wind process incorporates a dynamic wind direction scenario suited to the hexagonal turbine layout, where wind speed and turbulence remain governed by the OU process, but wind direction varies per episode. In future chapters, this process will be refered to as the fixed-direction-per-episode process. This variation is not random but is instead sampled proportionally to directions known to cause significant wake losses, as shown in figure 3.3. This strategic sampling addresses a common challenge in wind energy research: the scarcity of relevant data on wake effects due to the infrequency of specific wind directions in natural settings. By focusing on problematic directions, this approach ensures that data collection and subsequent learning are concentrated on scenarios where wake interaction is most detrimental to farm efficiency. This methodology is particularly beneficial for developing and testing wake steering strategies in complex layouts like the hexagonal configuration, where the central turbine's

influence varies significantly with wind direction. The ability to dynamically adjust the wind direction allows for a comprehensive evaluation of how well different control strategies mitigate wake effects and enhance overall farm performance under diverse conditions. This adaptive experimentation accelerates the learning process for control algorithms, enabling them to learn policies that can significantly boost the operational efficiency of wind farms. Such dynamic wind processes are crucial for advancing our capability to design robust, adaptable wind farm management systems that can optimize performance in the face of fluctuating and often unpredictable wind patterns.

3.3. FLORIS as a Simulator for Reinforcement Learning

To perform RL, many data samples are needed, as well as a simulator on which policies can be evaluated. To this end, FLORIS was used to construct a simulator. At each time step, the RL algorithm would be used to determine the yaw angles of the turbines, the wind processes above would be used to determine atmospheric conditions, and FLORIS would be used to calculate the individual power output of each turbine. This would be one individual cycle of an agent making observations, taking an actions, and evaluating the result.

Because REINFORCE is used, this process needs to be separated into episodes. To accomplish this, a wind process, simulating approximately one week of data, was generated and split into training episodes, each representing approximately one hour. This is presented in the figure below, which shows one long wind process chopped into episodes, with periodic evaluation episodes.

Figure 3.5: Wind process as episodes

At interval of 15 episodes, 5 evaluation episodes were conducted, each with completely distinct wind processes. Each evaluation episode is a longer episode, representing approximately 2 hours of data. The mean output over the 5 evalution episodes is what is used to evaluate the performance of each policy.

To effectively apply RL algorithms to the active wake control problem within wind farms, it's essential to reframe the issue in terms of discrete components such as time steps, states, actions, rewards, and transitions. To construct the state, simulated lidar masts were placed by the turbines to provide atmostpheric information about turbulence intensity, wind direction, and wind speed, at the turbine locations. These, along with the current yaw of the turbine were given as observations to each agent.

The action space was constructed relative to the incoming wind direction, and normalized on a range of $[-1, 1]$. In this setup, 0 means face the wind, and $(-)1$ means rotate towards the maximally (counter)clockwise direction. This setup aims for a more realistic problem formulation by integrating the heterogeneous data these tools provide, reflecting the varied environmental conditions across different parts of the wind farm.

Each time step in the RL framework is treated as having steady-state atmospheric conditions, which shift at the conclusion of the time step when the control actions are implemented, leading to a new state. This steady state assumption simplifies the modeling but still accommodates the dynamic nature of wind conditions by allowing for changes at each discrete time interval. The state space within this RL setup is defined by the observable parameters at any given time step, which in practice would include the current yaws of all turbines, wind speed, direction, and turbulence intensity, each measured at specific locations within the farm using tools like nacelle-mounted lidar. While FLORIS simulations

can provide a comprehensive representation of these conditions, only the data captured by the actual measurement tools are used to define the state in the RL model. This approach ensures that the state vector used in simulations reflects practical, observable data, enhancing the realism and applicability of the RL algorithms to real-world wind farm operational strategies. For accurate modelling, the power curve of a National Renewable Energy Laboratory 5 MW turbine was used [21] in all simulations.

3.4. Influence Heuristics

Heuristics in RL serve as simplified strategies or rules of thumb that guide decision-making processes, particularly in complex environments where exhaustive search and computation are impractical. These heuristic methods allow RL algorithms to make more efficient and effective choices by incorporating domain-specific knowledge and approximations. In this context, the following heuristic is proposed for evaluating the influence among turbines in a wind farm setup. Specifically, the heuristic posits that one turbine's influence on another is significant if it is located within a distance equivalent to six rotor diameters and if the angular deviation between the two turbines, relative to the wind direction, is less than 5 degrees. This heuristic simplifies the assessment of influence relationships, which are critical for optimizing turbine policies and maximizing energy output, by providing a practical and computationally efficient criterion for influence estimation.

In figure 3.6 the exact rules that will be used can be seen. In the leftmost diagram, an influence relationship is demonstrated., since the downwind turbine is within the distance and angle thresholds. The second picture describes the concept of influence levels, which will be used to group turbines in later chapters. A turbine is level $n+1$ if it influences a turbine of level n. Turbines which do not influence any other turbines are level 0. Lastly, the influence matrix on the right is a tool which shows which turbines influence each other.

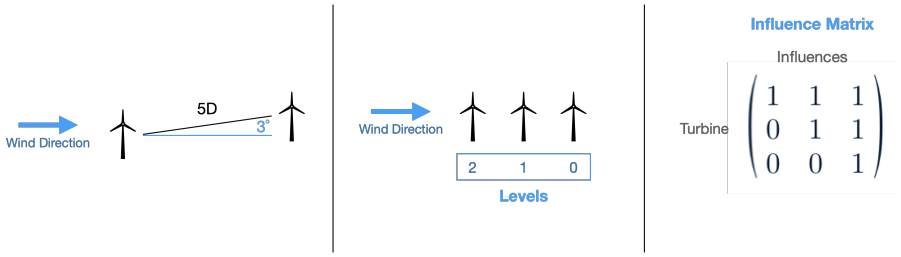

Figure 3.6: Influence Heuristics

4

Difference Rewards

This chapter introduces the concept of difference rewards as a solution to the credit assignment problem for AWC. By assigning each agent a share of the reward that reflects their individual contribution relative to a baseline, difference rewards provide a clearer understanding of each agent's impact. This method allows for modeling the wake relationships among wind turbines through the reward function, as it accurately captures the influence of one turbine's actions on others. Consequently, employing difference rewards can lead to improved coordination and performance in multi-agent reinforcement learning scenarios, ultimately optimizing wind farm operations.

4.1. Background

Cooperative MARL is an evolving field that focuses on developing strategies where multiple agents learn to work together towards a common goal. This paradigm extends the traditional reinforcement learning framework to scenarios involving several decision-makers, each with the ability to observe the environment, make decisions, and learn from the consequences of their actions. The cooperation among agents is crucial in complex environments where the actions of individual agents are interdependent, and the collective behavior determines the overall system performance. However, facilitating effective cooperation in MARL poses unique challenges, primarily due to the dynamics of learning shared strategies and ensuring that individual agent actions contribute positively to the group's objective.

One of the main issues in implementing cooperative MARL is the use of a shared reward for all agents, which often leads to the credit assignment problem [26]. When all agents receive the same reward, it becomes difficult to discern which actions by which agents contributed to the outcome. This ambiguity can hinder the learning process, as agents may either fail to recognize the value of their contribution or be rewarded for actions that do not positively affect, or even detract from, the group's goal. Consequently, agents might learn suboptimal policies that compromise the efficiency and effectiveness of the collective endeavor. The credit assignment problem thus represents a significant obstacle to achieving true cooperation and maximizing the performance of MARL systems.

Addressing this challenge, reward functions can be designed to encourage cooperation among agents. One innovative approach is the use of difference rewards, which aim to measure the impact of an individual agent's action by comparing the collective outcome with and without the agent's contribution. This method provides a more precise evaluation of each agent's role in the shared success, encouraging actions that genuinely benefit the collective goal. Difference rewards have been successfully applied in various domains, demonstrating their effectiveness in promoting cooperation and improving collective outcomes. For instance, in active wake control within wind farms, difference rewards [13, 12, 36] could significantly enhance the coordination among turbines to adjust their operational parameters in ways that collectively increase energy production while minimizing the adverse effects of wake interference. By accurately attributing the contribution of each turbine's adjustments to the overall energy yield, difference rewards offer a promising solution to optimize cooperative strategies in MARL applications, paving the way for more efficient multi-agent systems.

This method allocates individual credit to each agent by calculating the difference between the collective reward received and the hypothetical reward that would have been achieved had the agent taken a predefined "default action" instead [26, 3].

The formula for calculating this difference is:

$$\Delta R^i(a^i|s, a^{-i}) = R(s, a) - R(s, \langle a^{-i}, c^i \rangle) \tag{4.1}$$

where a_i represents the action taken by the agent i, a_{-i} denotes the collective actions of all other agents, and c_i is the default action for the agent i. This mechanism enables each agent to gauge its unique contribution to the group's performance, thereby providing a more granular insight into how individual actions affect collective outcomes.

Calculating difference rewards is not always possible. The calculation requires access to a complete reward function or a simulator that can be reset to test different actions' outcomes Furthermore, it needs to be determined which action constitutes the "default action" for comparison purposes. These requirements, while potentially limiting the applicability of difference rewards in certain contexts, are met in AWC, making it a viable solution, given a steady state simulator. In addition, even in a fluid dynamics based simulator, which does not satisfy either requirement, the reward function can be learned through supervised learning [10]. By applying difference rewards in such settings, it becomes possible to finely tune the contributions of individual turbines (or agents) towards optimizing the overall efficiency of the wind farm, enhancing energy production while mitigating the complexities of the multi-agent credit assignment problem.

4.2. Method

These experiments will use the triangle wind farm layout and the fixed-direction wind process described in the experiment setup chapter. It uses the triangle layout because the wake effect will increase as more turbines are in a rows. Therefore, this layout shows how severe the wake effect is at different depths. The fixed direction wind process is used in order to allow turbines to learn their effect through the rewards. This maintains a consistent environment in which turbines can optimise their policies.

With the aim of improving cooperation and optimizing performance in MARL for AWC, several reward shaping methods were tested. The reward shaping strategies, designed to solve the credit assignment problem and foster agent cooperation, are assessed by the speed of convergence, and the energy output once converged. We begin by establishing a standard cooperative MARL setup as our baseline, against which we benchmark the performance of more advanced reward strategies. This baseline method sees every agent running the REINFORCE algorithm independently, with the power output of the entire farm being used as the reward. The second method subtracts the farm's total power under a default action scenario, which is where all turbines face the wind, from the total power generated using the learned policy. The next method is DrREINFORCE, which is applied to the problem using the individual turbine facing the wind as the default action. Finally a heuristic based difference rewards based on the spatial dynamics between turbines, is tested. Each method progressively builds on the last, introducing complexity in how difference rewards are calculated and applied. This progression aims to not only validate the efficacy of difference rewards in a cooperative context but also explore the potential for tailored approaches that account for the unique interactions between turbines, thereby optimizing the learning and operational strategies within a wind farm environment.

In our investigation, we explore three distinct implementations of reward shaping to optimize cooperative behavior and energy production in wind farms through MARL, which are compared to every agent independently running REINFORCE. The results are also compared to the default policy used in practice, which is to always face the wind. The policy of always facing the wind is referred to as the "naive" strategy. The initial approach establishes a baseline by employing the total power production of the wind farm as the collective reward shared by each agent. This will be referred to as multi agent (MA) REINFORCE. This baseline is crucial for comparison, serving as a reference point to gauge the effectiveness of more sophisticated difference reward strategies subsequently tested. The use of total farm power as a shared reward inherently promotes a level of cooperation among agents, as it aligns their objectives towards a common goal—maximizing the overall energy output. However,

this method also allows for the understanding of dynamics of reward sharing and its impact on the learning process. Specifically, it provides insights into how agents, through the REINFORCE algorithm, eventually converge towards strategies that enhance collective performance. This approach allows us to observe the initial cooperative behavior induced by sharing rewards but also highlights the potential and challenges of scaling and refining reward mechanisms to foster more complex cooperative interactions and efficient learning outcomes in MARL environments.

$$R^i = R(s, a)$$

The second method we propose introduces a version of REINFORCE with baseline. Specifically, we calculate the reward for an agent by subtracting the total power production of the farm—under the condition that all agents take this default action—from the actual total power production achieved with the agents' chosen actions. This method allows the turbines to more clearly see their effect on the group output, and it has the added benefit of normalising for wind speed, which can confound learning in the baseline scenario.

Incorporating the REINFORCE algorithm with a baseline in this context brings additional benefits, particularly in terms of reducing variance in the reward signals received by each agent. The variance reduction is crucial for stabilizing the learning process, making the agents' policy gradient ascent more efficient and leading to faster convergence towards optimal strategies. This is because the baseline helps to normalize the rewards agents receive, making it easier to identify the effectiveness of specific actions amidst the noisy environments typical of wind farms. By effectively distinguishing between high and low-impact actions, agents can adjust their strategies more precisely, enhancing the collective energy production with a clearer understanding of individual contributions.

$$R^i = R(s, a) - R(s, c)$$

The introduction of DrREINFORCE as our third method marks is a further advancement in our exploration of difference rewards within the domain of MARL for AWC, because it rewards each agent individually instead of using the farm wide reward. This approach addresses the credit assignment problem by calculating the difference reward for an agent, i, through the subtraction of the joint reward, if agent i were to select a predefined default action, from the joint reward achieved with the action actually selected by the agent, a_i. This calculation method calculates the individual individual contribution of each agent to the collective outcome but also works well with the convergence guarantees provided by the REINFORCE algorithm. By directly associating the impact of an agent's specific action with the overall performance, DrREINFORCE ensures a more accurate and fair evaluation of agent contributions. Agents are thereby lead to iteratively improve their strategies, for the benefit of the entire farm. This leads to improved cooperative behaviors and optimized energy production, while avoiding the complexity in the dynamic and interconnected environment.

$$R^i = R(s, a) - R(s, \langle a^{-i}, c^i \rangle)$$

The fourth method in our exploration introduces a heuristic-based approach to calculating difference rewards, advancing the application of difference rewards in MARL for AWC. This method innovates by integrating a heuristic to assess the influence of each turbine's actions on others, taking into account the spatial dynamics such as the distance and angle between turbines relative to the wind direction. This allows for a more granular calculation of difference rewards, similar to the DrREINFORCE methodology but with a modification: the rewards are adjusted based on the actions' impacts on a subset of turbines deemed relevant through the heuristic. This approach also enables a targeted evaluation of an agent's contribution, focusing on the immediate network of influence within the wind farm.

By employing this heuristic-driven method, we aim to address the complexities of wake interaction and aerodynamic dependencies in a densely packed wind farm, where the actions of one turbine can have a cascading effect on the performance of others. This method aims to accurately attribute contributions to the overall energy production and to lead to a cooperative operational policy. The last method will use a heuristic for determining influence, based on the distance and angle between turbines given the

wind direction, and calculate difference rewards in the same way as DrREINFORCE, except differencing according to a subset of the relevant turbines:

$$R^i = \sum_{t \in F(s,i)} R^t(s,a) - R^t(s, \langle a^{-i}, c^i \rangle)$$

Where F is a function which determines whether a turbine has been influenced by i given the state. In for our research F is defined by the influence heuristics described in the Experimental Setup chapter, where one turbine is said to influence another if it is within six rotor diameters and the angle between them is less than 5°. This method also enables parallel calculation of difference rewards across levels, leading to improvements in training time.

4.3. Results

The application of difference rewards strategies in AWC has produced the following outcomes, as depicted in the provided graph, which tracks the energy production over a span of an evaluation episode for five distinct testing conditions: Naive, MA REINFORCE, REINFORCE with baseline, DrREINFORCE, and Influence Rewards.

Naive - Every agent faces the wind.

MA REINFORCE - Each agent runs REINFORCE with collective rewards.

REINFORCE with baseline - Each agent runs REINFORCE with a baseline of the reward if all agents took the default action receives the difference between the collective reward given the selected action, and the collective reward given all agents take the default action.

DrREINFORCE - Strategy from the Difference rewards policy gradients paper.

Influence Rewards - Select influenced turbines based on the influence matrix, and difference rewards based on the selected turbines.

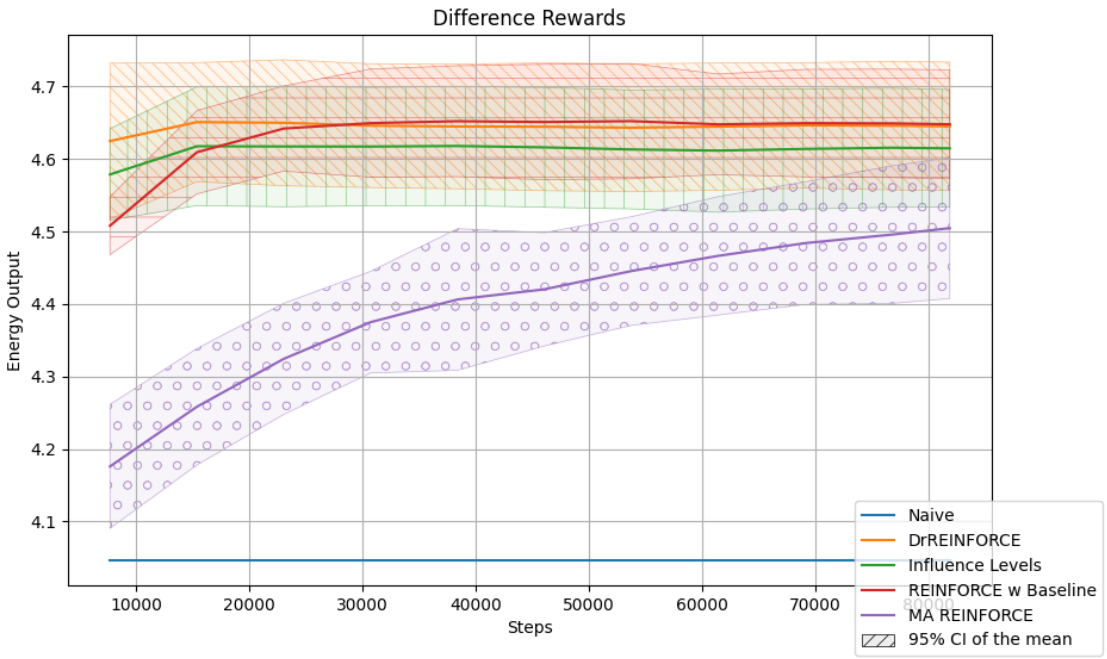

Figure 4.1: Energy Production per evaluation episode

As shown in 4.1, all policy gradient strategies converged to a similar reward over time, and they all outperformed the naive strategy, shown in blue. The reward modification strategies converged to

this higher value in fewer steps, indicating an improvement in learning due to reward differencing. The graph exhibits the mean energy output along with the 95% confidence intervals, showcasing the volatility and the stability of each strategy.

The policy gradients with no modifications showed slower learning than all reward modification strategies. This test case clearly underperforms compared to the strategies utilizing reward modification. Furthermore, there is a broading confidence interval, indicating a high degree of variability in performance. This higher variability is characteristic of REINFORCE without modification.

The multi agent policy gradients with baseline showed a similar pattern of improvement to the difference rewards and influence rewards. Compared to policy gradients with no modification, learning converged quicker. Furthermore, variance was reduced compared to policy gradients with no modification. Lastly, compared to the difference rewards strategies, the computational costs were lowered as shown in the by the time costs in figure 4.3.

DrREINFORCE and Influence Rewards, where the difference reward was based on the total power production of the farm minus the power with the agent in question taking the default action, demonstrated considerable improvements in energy output. The energy output converged quickly, and the confidence intervals remained relatively tight, suggesting consistent performance across different training episodes.

Since all the difference rewards strategies lead to convergence before the first evaluation, the behaviour of the algorithms on a shorter time frame is shown.

Figure 4.2: Energy Production over a limited optimisation time

The REINFORCE baseline strategy, represented by the red line and shaded area, shows slower learning compared to the other strategies.

The REINFORCE with baseline strategy, represented in blue, displays a learning curve that lies between the baseline REINFORCE and the other two reward modification strategies. It converges more quickly than the baseline and has a narrower confidence interval, suggesting more consistent performance than the baseline REINFORCE but slightly less so than DrREINFORCE and Influence Rewards.

DrREINFORCE and Influence Rewards strategies show a marked improvement in energy production compared to the baseline REINFORCE. They both converge faster to higher energy production values,

but they both have wider confidence intervals in the beginning. However, as shown in 4.1 the confidence interval narrows as the experiment progresses.

The reward modification strategies (DrREINFORCE and Influence Rewards) demonstrate superior performance both in terms of faster convergence to higher energy production and in maintaining consistency across learning episodes. REINFORCE with baseline also shows improved performance over the baseline but not as pronounced as the other two strategies. The analysis indicates that modifying the reward function can lead to more efficient and stable learning in reinforcement learning tasks.

The results corroborate the hypothesis that difference rewards can significantly enhance the performance of AWC systems by providing a more targeted approach to credit assignment among agents. The REINFORCE with baseline, and influence rewards appear to be the most promising candidates for future exploration due to its quick convergence and computational efficiency, after the initial learning period. This suggests that an understanding of inter-turbine influences is critical for optimizing wind farm efficiency. In contrast, the naive approach without strategic reward structuring serves as a clear baseline, underscoring the benefit of reward mechanisms in complex, cooperative multi-agent environments.

4.3.1. Computational Costs

Employing a heuristic based on influence, the influence rewards REINFORCE showed quick convergence to the optimal policy, just as DrREINFORCE. These different reward differencing strategies are roughly equally expensive in terms of computational complexity, as shown in the following figure. In order to parallelize influence rewards, the differencing is simply done once for each influence level, rather than once for each agent. This should yield equivalent results since no turbines in the same level influence one another. This should reduce the training time, since the simulator will only have to be reset, and the wakes only have to be recomputed, once for each level, rather than once for each turbine. This reduces the number of wake calculations from six to three in the triangle layout.

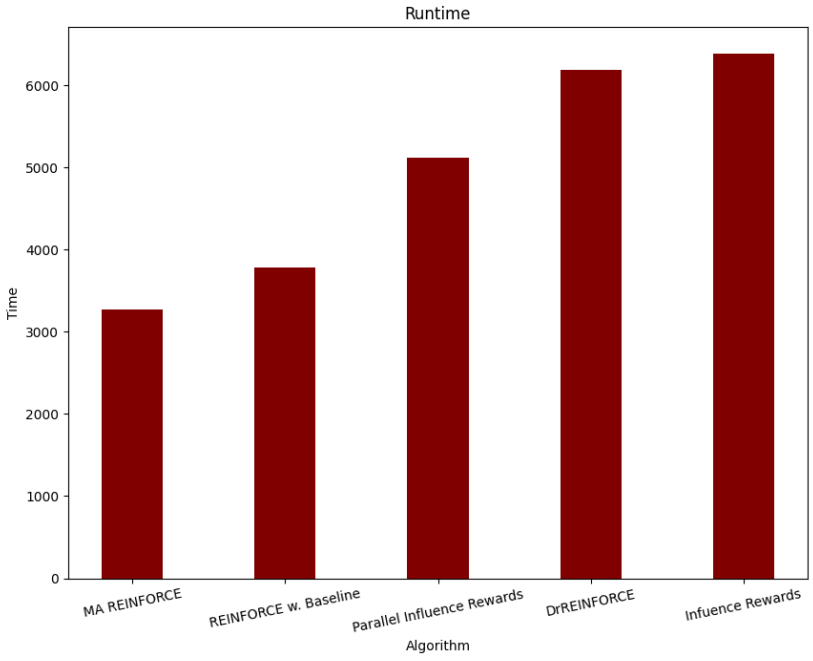

Figure 4.3: Experiment run time by solution

The figure shows that the hypothesis is correct, and differencing at each level speeds up the experiment. In practice this results in reducing training time by approximately 16%. The gains will be greater the larger the farm, since the number of levels increases slower than the number of turbines.

4.4. Conclusion

In conclusion, this chapter has demonstrated the significant advantages of difference rewards in optimizing the performance of cooperative MARL for AWC in wind farms. Through an examination of various implementations, including REINFORCE with and without baselines, DrREINFORCE, and heuristic-driven Influence Rewards, it was clear that difference rewards address the critical challenge of the credit assignment problem effectively. By modeling the distinct contributions of individual turbines, these rewards expedite the learning process, enabling more precise and efficient optimization of energy output. Difference rewards created a deeper understanding of the interdependent dynamics within the wind farm by quantifying the specific impact of each agent's actions in relation to a baseline scenario. This lead to quicker convergence on optimal strategies, as demonstrated by the steeper learning curves and convergence to higher energy output levels.

The comparative analysis of the reward structures demonstrated the superiority of difference rewards in promoting effective cooperation among turbines. By aligning each agent's incentives with the collective goal of maximizing energy production, difference rewards can enhance operational efficiency. Future research should continue to refine these reward mechanisms, and explore supervised learning of reward functions for use in more advanced fluid dynamics based simulators, such as large-eddy simulation.

5

Training Regimes

In this chapter, I introduce training regimes as a method to address the moving targets problem in multi-agent systems. This problem arises from the shifting policies of other agents, which cause underlying changes in the environment. Training regimes can aid in mitigating these changes by keeping the environment, particularly the behaviors of other agents, as stable as possible. This is done with the aim of facilitating more consistent and effective optimization, enabling agents to learn and perform better without the interference of constantly changing dynamics.

5.1. Background

In the landscape of MARL, the development of training methodologies can play a role in enhancing the performance and stability of learning agents [35, 34]. Among these methodologies, staged training, ensemble training, and various other training regimes have been shown to contribute significantly to the efficiency of learning processes.

Training regimes for MARL encompass a wide array of strategies designed to optimize the learning process. These include, but are not limited to, curriculum learning, where tasks are progressively made more difficult; reward shaping, which involves modifying the reward function to guide agents towards desired behaviors more quickly; and experience replay, where agents revisit past experiences to reinforce learning. The choice of training regime depends on the specific challenges of the environment, the desired outcomes, and the inherent capabilities of the learning agents. Effective training regimes are tailored to encourage exploration, ensuring that agents do not become trapped in suboptimal policy loops, and exploitation, allowing agents to refine their strategies to maximize rewards.

Staged training refers to a sequential learning approach where the training process is divided into distinct phases or stages, each with specific learning objectives and complexity levels. This method is beneficial MARL contexts, where navigating complexities of the environment can lead to learning not picking up, or to learning a suboptimal policy. Staged training can facilitate a smoother learning curve, preventing agents from being hindered by the complexity of tasks beyond their capabilities [27]. Staged learning can be applied to AWC by training different agents at different times to ensure that they are learning in the simplest possible environment.

The application of these training methodologies in MARL is driven by the need to address the challenges presented by complex environments. In RL, where the focus is on optimizing the decisions of a single agent, staged and ensemble training can significantly speed up the learning process, helping the agent to navigate complex state-action spaces more efficiently [41]. In the more dynamic and complex environment of MARL, these training methodologies are useful for managing the additional layer of complexity introduced by agent interactions. They facilitate not just the learning of effective individual strategies but also the development of mechanisms for cooperation among agents.

In MARL, the moving targets problem poses a significant challenge, wherein the underlying environment evolves dynamically while learning is underway. This problem is particularly pronounced in active

wake control, where the environment not only changes inherently but also in response to the policies adopted by neighboring agents. To mitigate these complexities, various staged training methodologies have been developed, and tested.

One effective approach involves interleaving learning between agents, as suggested by prior research [35]. A straightforward implementation of this strategy is to training of all agents sequentially, such that only one agent is performing gradient updates at any given time. By doing so, each agent operates in a simplified environment, thereby facilitating more stable learning. Moreover, agents can be grouped based on logical tasks to encourage cooperative behavior [34].

The second approach also cycles through agents, training each one individually, while interleaving episodes where all agents are trained. The underlying hypothesis is that when a single agent is undergoing training, it benefits from a more stable learning environment, leading to more efficient learning, but the benfits of training as a whole are also retained. By employing staged training techniques, MARL systems can navigate the challenges posed by dynamic environments and moving targets, thereby facilitating improved learning and coordination among agents in complex scenarios such as active wake control.

5.2. Method

The experiments will use the triangle wind farm layout and the fixed-direction wind process. The triangle layout is used because a training regime, in order to be effective, must be able to accomodate different levels of wake induced losses. The fixed wind process is used because these different levels of wake induced losses should be maintained accross the duration of the experiment.

We explore and evaluate three distinct approaches to staged training within the context of AWC, in addition to REINFORCE. These methods are designed to improve the efficiency and output of each turbine, as well as the overall farm, through stabilising the learning environment. A variety of different numbers of iterations before switching training turbine were experimented with, to find the best for each strategy.

The first approach focuses on individual turbine optimization. In this method, each turbine is trained independently to maximize the performance of the wind farm. This can be seen in the pseudocode below 6, where the agent is selected based on the episode. The EpisodeGroupSize variable shows how many times to train an agent before switching to the next one. This also allows for repeating the process as many times as desired. During this phase, the learning process for all other turbines is temporarily halted, effectively isolating each turbine's learning process. This isolation is achieved by not performing gradient update steps for any turbines other than the one currently being trained, ensuring that each turbine's optimization does not influence the others.

Algorithm 2 One By One Training

```
Agents = {A1, A2, ...}
EpisodeGroupSize = x
for Episode in 1..NumEpisodes do
    Observations, Rewards, Actions = RUNEPISODE()
    AgentIndex = (episode - 1) // EpisodeGroupSize
    Agents[AgentIndex].LEARN(Observations, Rewards, Actions)
end for
```

The second strategy introduces an alternating training regime, where the focus shifts between optimizing the performance of a single turbine and then the entire farm collectively. This method aims to balance the benefits of individual turbine optimization with the advantages of joint farm optimization, allowing the individual agent to learn its influence on the whole, and for the whole to learn collectively. This is shown by the if-statement in the pseudocode below. It shows that for the first half of each agent's training period the whole group is trained, and for the second half, only the individual agent is trained. By periodically shifting focus, this approach seeks to optimize performance of individual turbines with the overall efficiency of the wind farm.

The third method adopts a hierarchical training approach, grouping turbines based on the number

Algorithm 3 Alternating Training One and All

```
Agents = {A1, A2, ...}
EpisodeGroupSize = n
for Episode in 1..NumEpisodes do
    Observations, Rewards, Actions = RUNEPISODE()
    if Episode mod EpisodeGroupSize < (EpisodeGroupSize / 2) then
        for Agent in Agents do
            Agent.LEARN(Observations, Rewards, Actions)
        end for
    else
        AgentIndex = (Episode - 1) // EpisodeGroupSize
        Agents[AgentIndex].LEARN(Observations, Rewards, Actions)
    end if
end for
```

of turbines they are influencing. Turbines within the same level are trained together as a cohort, and the focus of training shifts between these levels at regular intervals. This is handled by the *CALCULATE_INFLUENCE_RELATIONS* method, which returns a map mapping each influence level to a list of agents. This is done using the method described in the Experimental Setup chapter. This strategy allows for a layered optimization process, where groups can be optimised such that they learn in a steady environment.

Algorithm 4 Training by Level

```
LevelToAgents = CALCULATE_INFLUENCE_RELATIONS()
EpisodeLevelSize = n
for Episode in 1..NumEpisodes do
    Observations, Rewards, Actions = RUNEPISODE()
    Level = (Episode - 1) // EpisodeLevelSize
    for Agent in LevelToAgents[Level] do
        Agent.LEARN(Observations, Rewards, Actions)
    end for
end for
```

These three training methods are evaluated against the baseline approach of using unmodified REINFORCE. By comparing these three staged training strategies against the naive strategy and standard REINFORCE solutions, the experiments aim to identify the most effective methods for optimizing wind turbine performance, taking into account the interactions between turbines within a wind farm.

5.3. Results

The use of different training regimes with REINFORCE for AWC produced the outcomes depicted in 5.1, which shows the total energy produced during one evaluation episode. The experiment conditions are as follows:

Naive - Each agent faces the wind.

MA REINFORCE - Each agent runs REINFORCE with no modifications.

One One - Each agent runs REINFORCE, but the agents are trained one by one. Each agent is trained for five episodes, and the turbine which influences two downstream turbines is trained first, followed by the turbines which influence one turbine earch, followed by the last row of turbines.

One All - Each agent runs REINFORCE, and the training is alternated between training an individual turbine and training the ensemble as a whole. The training order is the same as in the One One experiment

By Level - Each agent runs REINFORCE, and training is alternated between levels, with each level being trained as a group. The order is by the number of turbines influenced.

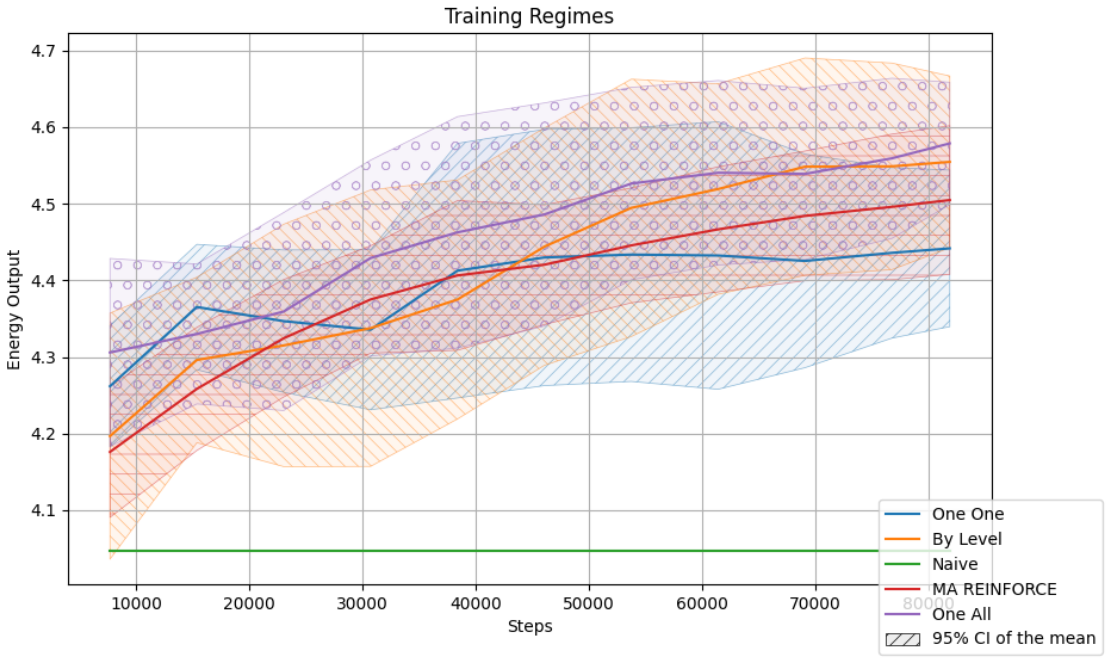

Figure 5.1: Energy Production of Staged Training Methods

As shown in 5.1, performance was very similar between the strategies, since none of them fully converged in the given time period. The training regimes lead only lead to small differences in performance. Furthermore, they did not all improve the performance, as in the case of training one turbine at a time, which diminished the performance.

REINORCE with no modifications showed an improvement in policy, but it underperforms both the test case which alternates between the group and the individual, as well as training in groups of the same influence level. It did perform better than training turbines one by one, suggesting that there is benefit to training as a group.

Training one agent at a time lead to lower performance than the baseline REINFORCE algorithm. This could be caused by each agent simply having fewer iterations to update their policy, and therefore, letting the training proceed would lead to better performance. However, this may not be the case, since the inital performance started off better than the baseline, and the confidence iterval is much wider, suggesting slower and more uncertain improvement.

Training by level lead to an improvement in performance over the baseline REINFORCE algorithm. This confirms that making the environment more stationary has a positive influence on learning. The performance improvement over the baseline is likely due to this more stationary environment, combined with the greater number of gradient descent steps each agent takes in comparison to training one agent at a time.

Alternating training between the individual agent and the ensemble lead to the best performance, and to the narrowest confidence interval. This suggests that the is benefit to both training as a group and training as an individual leads to improvement. This is surprising considering that training each agent individually lead to worse performance overall.

The results suggest the hypothesis is ambiguous, and that creating training regimes can lead to both better or worse performance. It should be noted, however, that due to the overlapping confidence intervals, these claims cannot accepted wholesale because the results may change with further experimentation.

Overall, it can be said that stabilising the environment using training regimes may lead to improved performance, but this is not fully confirmed by the results.

One question that arrises from the results is whether the order of training agents can make a difference in the performance, similar to the grouping of agents being trained.

5.3.1. Training Order

Since the goal is to minimise the variability in the environment, such that each agent learns in an environment which is as stable as possible, leading to improved learning. To this end, the order in which the agents are trained may make a difference in addition to the grouping of agents being trained. By training in order of descending influence level, which is to train the most upwind turbine first, could stabilize the environment for the downwind turbines, such that they only learn with higher wind speeds. Training the downwind turbines first could improve learning because the difference in reward which the upwind turbine would see as it yaws would be greater. However, it is possible that the upwind turbine yawing, leading to higher wind speeds for the downwind turbines, would lead to decreased performance of the downwind turbines since they will then be exposed to previously unseen observations.

To test whether the order of training impacts performance, the same experiments were conducted, but each method was compared to itself with the order of training differing. Ascending means the turbines were trained in order of ascending influence level, therefore, from most downwind to most upwind. Descending is the opposite, from most upwind to most downwind.

As shown in figure 5.2, in all cases, training in order of descending influence level, which is to say, starting upwind and progressing downwind, lead to a narrower confidence interval. This confirms the hypothesis that training the most influential agents first leads to a stable environment in which the less influential agents learn. This, in turn, leads to better overall stability. Overall, the mean performance of the training regimes was not changed by the manipulating the training order. However, when Training one by one there was a larger difference in mean evergy output than for the others, and training in an ascending order of influence lead to higher performance. This was not expected, and could be caused by the most influential turbines finding a suboptimal policy first, and the ascending case, there is a more clear reward signal due to the downwind turbines already producing maximal power.

5.4. Conclusion

In this chapter, we explored various staged training strategies within the context of AWC using MARL. The methodologies tested include training by level, which enhances stability by allowing simultaneous optimization of multiple turbines, and alternating training, which focuses sequentially on individual turbines and the entire ensemble. These approaches were designed to mitigate the dynamic complexities of the learning environment, thereby mitigating the moving targets problem.

Our findings indicate that training regimes can both stabilizes or disrupt the learning process. In the best case, they can lead to a more controlled environment where groups of turbines with similar influences are optimized together. This method not only ensured that the learning environment is less volatile but also allows for more comprehensive and simultaneous optimization efforts across multiple turbines, in contrast to training each turbine individually. On the other hand, the strategy of alternating between training one agent and the whole ensemble slightly enhanced both performance and stability. The method balanced individual and collective learning needs, enabling turbines to refine their strategies in relation to the collective goals of the wind farm.

The training order was determined to significantly contribute to stability. We conclude that training in order of descending influence creates a more table environment for downwind turbines. This, in turn, mitigates complexity, and lead to a more stable environment in which the downwind turbines optimise their policy.

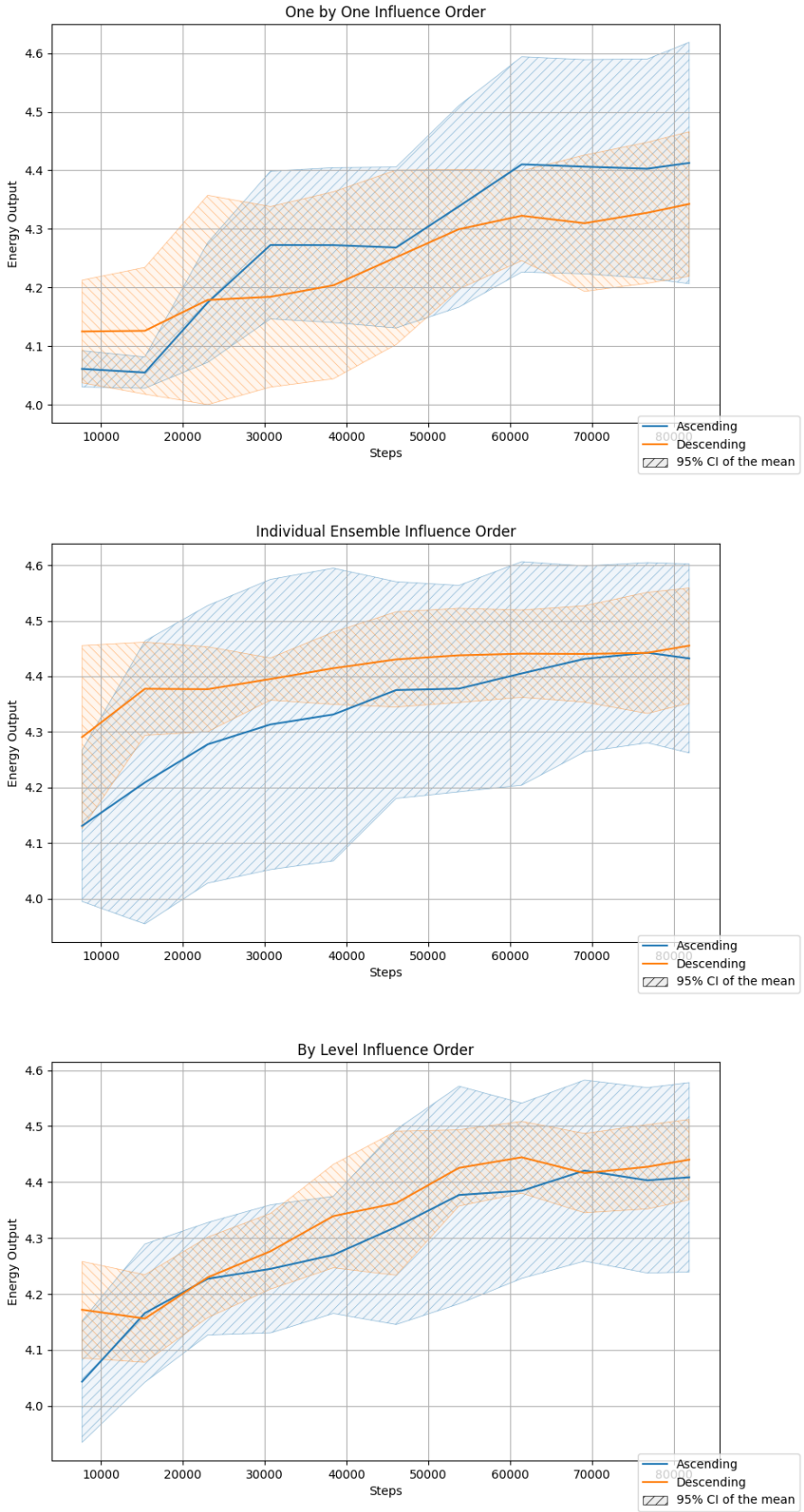

Figure 5.2: Energy Production of staged training methods by training order

6

Role-Based Reinforcement Learning

Since difference reward lead to fast convergence to a good policy in the wind tunnel setup, we take a step towards a more realistic scenario by changing to the fixed-direction-per-episode wind process. This increases complexity since the main environmental factor which determines the optimal policy is the wind direction. Therefore, this chapter will explore role-based reinforcement learning as a strategy to manage the increased complexity of wind farm optimization in environments with dynamic wind direction. Since turbines will alternately be upwind and downwind of others, they inherently play multiple roles. By defining agents as roles rather than individual turbines, each agent can specialize in its designated role. This specialization aims to simplify the learning task, reducing the complexity each agent faces and enhancing the efficiency and effectiveness of the learning process, even as wind conditions change.

6.1. Background

Role-based MARL integrates the complexity of managing multiple agents with specific, dedicated roles aimed at enhancing collective performance. This approach is helpful in dynamic environments such as wind farms, where each turbine can be conceptualized as an agent with a unique role, which changes based on environmental factors. This segment delves into the foundational strategies and implementations of role-based MARL and its application in active wake control, drawing on several studies.

In role-based MARL, agents are assigned distinct roles or tasks, optimizing their individual contributions while considering the overall system's goals. This specificity in roles allows for a more focused learning process, where agents specialize in tasks that maximize the collective utility rather than pursuing overlapping objectives that could lead to suboptimal performance. Li et al. [23] Emphasize that introducing diversity in roles and behaviors through information-theoretical regularization significantly enriches the exploration space and enhances coordination among agents. This suggests that such diversity in roles could be beneficial for managing wind farms where turbines need to adopt different strategies based on their position relative to others and the prevailing wind conditions.

Active wake control in wind farms represents a practical application of role-based MARL, where each turbine, or agent, must adjust its operational parameters to optimize both its power output and the overall efficiency of the wind farm. Jeon et al. [19] illustrate an advanced MARL framework where agents generate subgoals from an experience replay buffer, which effectively coordinates individual agents to maximize their collective output by learning subgoals. This is applicable to wind farms because the most upwind turbines will need to learn the same policy of yawing when their wakes would otherwise interfere with the production of downwind turbines. Furthermore, the introduction of intrinsic rewards for achieving these subgoals ensures that each turbine agent is motivated to align its operation within the broader objectives of the farm, thereby stabilizing the power production across varying atmospheric conditions.

Intergrating role-oriented learning into MARL can simplify learning in complex environments by limiting

the scope of each agent [49]. Focusing the training on learning roles that allow agents with similar responsibilities to share learning and specialize in sub-tasks. In previous research, role embedings have been derived by conditioning agent policies on role embeddings derived from local observations, as in Wang et al. [49] This has shown improved performance on the StarCraft II micromanagement benchmark, where the role-oriented MARL framework significantly enhances MARL efficacy by enabling adaptive policy sharing among agents with similar roles, demonstrating superior results compared to existing algorithms.

In Wang et al. [48] the dynamics between reward signals in the context of cooperative multi-agent systems are used to determine agent roles. By comparing different reward structures, the researchers aim to dissect how varying reward dependencies influence the learning behaviors of agents within the group. Their findings suggest that learning roles for each agent through the reward signal leads to more stable and cooperative behaviors in the long term. The paper emphasizes the importance of choosing an appropriate reward mechanism based on the specific requirements of the cooperative task to optimize overall system performance. In the context of AWC roles can be determined not by reward signal, but by geometric heuristics.

6.2. Method

For this experiment, the hexagon wind farm layout, and the fixed-direction-per-episode wind process. Since the objective of this experiment is to find a way to train agents to be able to optimise in an environment with a dynamic wind process, the fixed-direction-per-episode is used. The hexagon layout is used because it will allow for the most wake effects, and thereby the largest sample of data.

The wind farm consists of seven turbines arranged in a hexagonal pattern, with one turbine at the center and six surrounding it. This layout creates varied interaction patterns due to varying wind directions, which are useful for testing the efficacy of our role-based MARL approach. At the end of each episode, agents are assigned turbines which mach their role which the learn from. This is done using the influence matrix and influence levels described in Chapter 3.

The wind conditions are simulated using the second described wind process, where wind speed and turbulence are generated by an Ornstein-Uhlenbeck process, but the direction is sampled in episodes based on the directions that commonly cause wake losses. This approach ensures that the learning focuses on scenarios most detrimental to farm efficiency, thus providing a robust testing ground for the MARL strategies.

In this method, each agent does not reflect an individual turbine, but one of three roles that a turbine can be in. It can be the most upwind, affecting two turbines, it can affect one turbine, or it can be the most downwind, in which case it needs to face the find. Each agent takes a gradient ascent step at the end of each episode, using all the turbines which ocupy the relevant role. This approach tests the hypothesis that optimizing agents based on their task, rather than on their position will lead to agents which have to learn a simpler task

Algorithm 5 Level Agents

```
for Episode in 1..NumEpisodes do
    Observations, Rewards, Actions = RUNEPISODE()
    InfluenceLevels = CALCULATE_INFLUENCE_LEVELS()
    for AgentIndex in 1..NumTurbines do
        Agent = Agents[InfluenceLevels[AgentIndex]]
        Agent.LEARN(Observations, Rewards, Actions)
    end for
end for
```

A conventional REINFORCE algorithm is applied without role-based considerations or tailored learning strategies to optimize the turbines. This comparison helps to quantify the added value of role-specific adaptations in the MARL framework.

6.3. Results

The results from implementing a role-based MARL approach for AWC within a hexagonal wind farm layout under a dynamic wind process demonstrate significant improvements in turbine performance and overall energy output. The experiment compared the role-based MARL strategy, termed "Level Agents" since the roles were assigned based on the influence level of each agent, against a standard multi-agent REINFORCE approach and the naive strategy of always facing the wind.

The graph distinctly shows the performance trajectories of the three approaches. The Level Agents approach consistently outperformed the standard REINFORCE method throughout the experiment. Initially, both methods started with similar energy outputs, but as training progressed, the Level Agents strategy began to show a significant increase in energy production.

Figure 6.1: Energy Production per evaluation episode

Level Agents: This role-based approach yielded a notable upward trend in energy output, peaking higher than the standard approach. The confidence interval for this method is also narrower at the end of the experiment than at the beginning, indicating stable performance across different training epochs. The narrower confidence interval suggests that the role-based method leads to a more consistent and reliable optimization of turbine operations, reflecting the effectiveness of role-specific strategies in managing complex interactions within the wind farm.

Standard REINFORCE: In contrast, the standard REINFORCE algorithm exhibited a flat or slightly declining trend in energy output. The narrow confidence interval indicates a lower variance in performance, suggesting that this approach struggled to adapt to the dynamic wind conditions and complex turbine interactions effectively. However, REINFORCE did still outperform the naive strategy. This suggests that it learned a suboptimal policy before the first evaluation, and did not manage to improve further.

The role-based RL approach not only enhanced performance in terms of higher average energy output but also showed a narrowing confidence interval as the experiment progressed. This demonstrates the benefit of specialization in complex environments. Furthermore, the narrowing confidence interval is beneficial because stability is crucial for real-world applications where wind conditions are inherently variable and unpredictable. The role-based approach, by focusing on the specific influences and interactions of each turbine, was able to maintain a more consistent performance level, optimizing both

individual turbine output and overall farm efficiency.

These results underline the potential of role-based MARL in enhancing the effectiveness of AWC systems. By intelligently managing wake interactions and adapting turbine operations to changing conditions, such systems can significantly reduce wake-induced energy losses and increase the overall productivity of wind farms. The success of this approach in the experiment suggests promising applications in other multi-agent settings where dynamic interaction management is crucial. Howerever, given that the stated roles used for this experiment do not perfectly model the dynamics, the following subchapter will investigate the effect of creating even more specialized roles.

6.3.1. Further Differentiating Levels

By further examining the roles which the agents play, one can see that a turbine at influence level one is not always playing the same role. In one case it is the center turbine, which has the turbine which it influences blose behind it. In another case it corresponds to the yellow arrow in figure **??**, when the turbine is influencing a turbine further away. The following experiment investigates what happened when these roles are played by different agents. The hypothesis is that this will improve learning by allowing for further specialization.

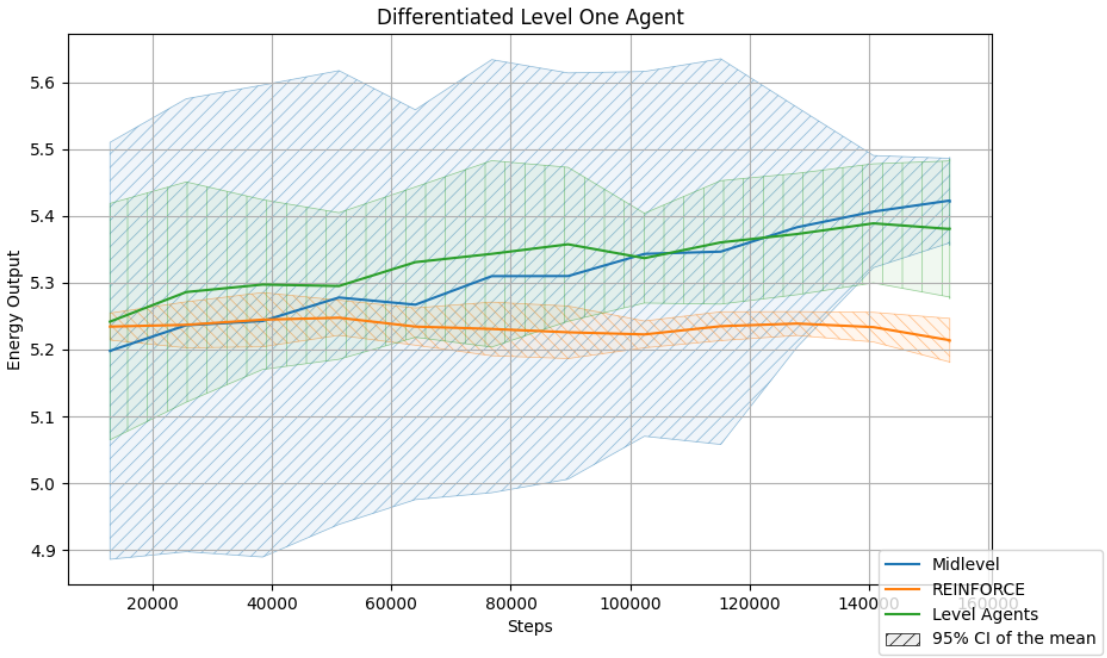

Figure 6.2: Energy Production with a Differentiated Level One Agent

The results show that in spite of one trial with an extremely poor initialisation, they all converge to a narrower confidence interval with a higher mean. The confidence intervals are, however overlapping, so it cannot be asserted that differentiating further produces strictly better results. However, the final confidence interval is much narrower, suggesting that even with poor initialisation, there is greater stability.

6.3.2. Domain Knowledge

To further improve the solutions, one may observe that the role of the most downwind turbines is always to face the wind, regardless of the configuration or policy of the upwind turbines. Fixing the rearmost turbines such that they always face the wind, which is their optimal policy, can lead to a more stable environment in which the other agents can learn their policy.

As shown, the agents have found a good policy far earlier, and the mean output of the final policy is slightly higher than for other policies.

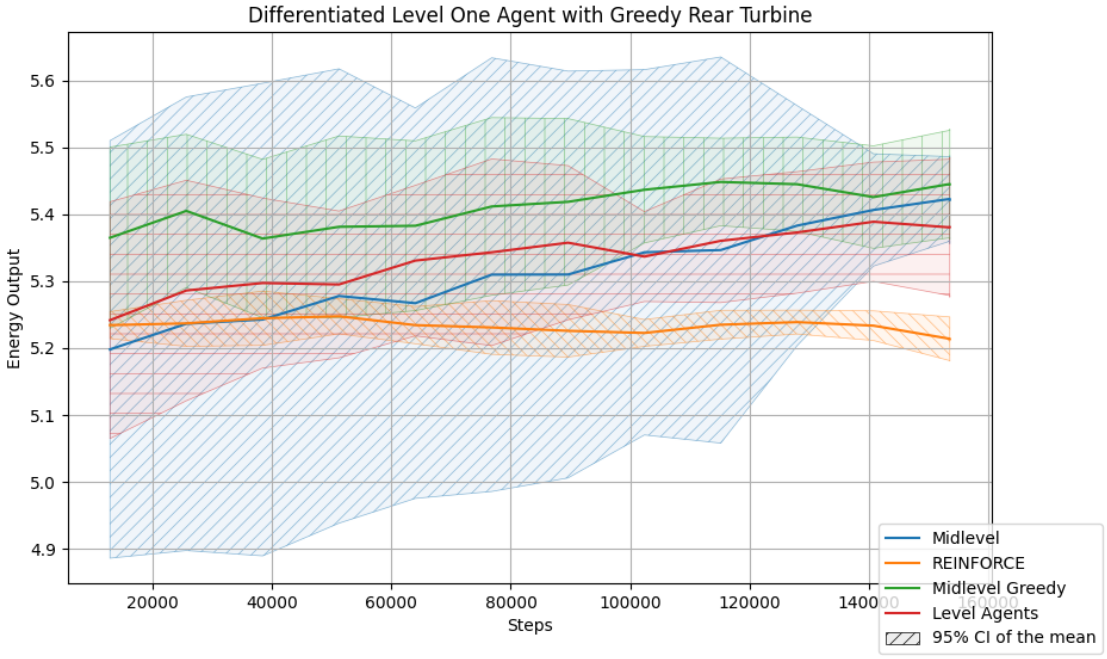

Figure 6.3: Energy Production with a Differentiated Level One Agent, with Domain Knowledge

6.4. Conclusion

In this chapter, we studied the efficacy of role-based MARL strategies in optimizing the performance of wind farms through active wake control. The role-based approach allows turbines to specialize in tasks that are critical for their specific positions to optimize power output while minimizing wake interference with neighboring turbines. This specialization is determined through the strategic definition of roles, which are based on heuristics for influence. This learning process leads to a more efficient adaptation to dynamic conditions, thereby increasing the overall energy production of the wind farm.

Our results strongly suggest that the more specialized the roles, the better the performance of the turbines. This is evident from the notable increase in energy output observed with the role-based MARL approach compared to a standard REINFORCE algorithm. Furthermore, it is clear that differentiating as much as possible leads improved results. In addition, by isolating roles, domain knowledge can be incorporated to simplify the problem even further, leading to increased stability. These findings highlight the potential of role-specific learning strategies in enhancing the operational stability and efficiency of complex systems like wind farms.

The success of this role-based approach underscores the importance of structured and specialized agent learning environments in managing complex interactions within dynamic systems. Future research could further explore the optimization of these role assignments as a supervised learning problem, which could lead to further generalisability and scalability.

7

Influence-Based State Abstraction

In this chapter we seek to determine whether state abstraction can be used to learn roles, in order to extend the benefit of role-based reinforcement learnign to scenarios where roles cannot be determined manually. Influence based abstraction (IBA) is a concept in the field of RL, which improves learning in RL by presenting abstracting away unnecessary information, and emphasising the most relevant information in observations. In the context of AWC, the objective is to present information about the influence relations in the observations in a better way than through the wind direction. This approach involves modeling the extent to which an upwind turbine affects those downstream, quantifying its "influence" on downstream turbines and presenting it in the observations By representing the number of turbines influenced by a given upwind turbine within observational data the decision is simplified, because the underlying network has a clear representation accross symmetries. Incorporating influence-based state abstraction into active wake control allows for more consistent performance accross scenarios.

7.1. Background

IBA is a technique designed to manage the complexities inherent in multiagent systems [11, 32], where multiple agents interact within a shared environment, influencing each other's decisions and outcomes. Developed as a method to simplify these interactions, IBA operates by segmenting a larger system into localized models that focus on the direct and indirect influences agents exert on each other. This approach is particularly useful in systems characterized by partial observability and decentralized decision-making, as it allows each agent to consider only the subset of the environment that significantly impacts its actions, rather than the entire system state [6]. By reducing the decision space to these relevant factors, IBA enhances computational efficiency and focuses resources on processing the most impactful data, which is essential in environments with limited computational capacities or where real-time decision-making is crucial.

The mechanism of IBA involves two primary processes: identifying the influential factors and abstracting these factors into a manageable model. Initially, the system identifies which aspects of the agent's environment or other agents' actions have a significant impact on its state and decisions [42]. It then abstracts these interactions into a simplified model that captures the essence of these influences without retaining as much complexity of the underlying system as possible [20]. This model delineates how agents' actions influence each other, often through a probabilistic framework that predicts the likelihood of certain outcomes based on specific actions. This method has been effectively applied in various domains, such as in decentralized control systems traffic intersection management, where understanding and predicting the influences of each unit's actions on others can significantly enhance cooperative task performance. Additionally, in the context of environmental management, such as AWC, IBA enables turbines to adjust their operations based on the predicted impact of their wake on downstream turbines, thereby optimizing overall efficiency and reducing energy losses due to suboptimal wake interference.

In the context of active wake control, influence-based abstraction can be particularly beneficial. Active wake control involves adjusting the operational parameters of wind turbines to minimize the negative

impact of wake effects, where the wind speed is reduced downstream of a turbine, affecting the performance of other turbines in its wake. By abstracting the influence of an upwind turbine on those downstream, the RL algorithm can model complex interactions more effectively. This is important since two separate wind direction can have the same optimal policy, and IBA can make these types of symmetrical relationships explicit [9] This abstraction allows for the development of optimized control strategies that consider the dynamic interdependencies between turbines, enhancing overall efficiency and energy output.

In summary, influence-based abstraction provides a framework for understanding and managing the dependencies and interactions within a multiagent system such as a wind farm. By abstracting the influence of individual turbines on others, it enables the development of sophisticated control strategies that enhance both the efficiency and reliability of wind energy production.

7.2. Method

For this experiment, the hexagon wind farm layout, and the fixed-direction-per-episode wind process. Since the objective of this experiment is to find a way to train agents t be able to optimise in an environment with a dynamic wind process, the fixed-direction-per-episode is used. The hexagon layout is used because it will allow for the most wake effects, and thereby the largest sample of data.

AWC is only relevant in the situation where at least one turbine is downstream of another. In large modern wind farms, this will frequently be the case, but for the smaller test cases used so far, there is not enough data with wake induced losses for the agents to learn. The first step to get more data is to change the wind farm layout to a hexagon, as shown in 3.2. The reason the hexagon layout is chosen is because the outer turbines can influence zero, one, or two turbines, and the inner turbine can influence any of the turbines. Furthermore, each outer turbine has multiple ways it can effect a single turbine. This is part of the reason for using IBA; two different states can be require the same policy, and that can be shown through the observations.

To perform IBA, the wind direction, which is the primary determinent of whether a turbine should yaw, should be abstracted. To accomplish this, the heurstic for calculating a turbine's influence level, which was shown in chapter 3, is used. The influence level is how many turbines are influenced by the given turbine. This is shown in the $CALCULATE\_INFLUENCE\_LEVELS$ method, which assigns each turbine to a level. The influence level is given as an obervation to each turbine in a one hot encoding, to make influence relationships and symmetries accross wind directions explicit. This is chown below as the concatenation of the observations and the influence levels into a new variable, which the agent uses to learn.

Algorithm 6 Observations with IBA

```
Agents = {A1, A2, ...}
for episode in 1..NumEpisodes do
    Observations, Rewards, Actions = RUNEPISODE()
    InfluenceLevels = CALCULATE_INFLUENCE_LEVELS()
    NewObservations = CONCATENATE(Observations, InfluenceLevels)
    for Agent in Agents do
        Agent.LEARN(NewObservations, Rewards, Actions)
    end for
end for
```

To test the use of IBA for AWC, three scenarios have been selected to evaluate how mcu information is gained by the addition of influence levels to the observations. The first is the standard REINFORCE algorithm with each turbine being a separate agent. This acts as a baseline experiment, against which the performance can be measured The second experiment modifies the observations by adding the influence levels as a observations in addition to the standard turbulence, yaw angle, wind direction, and wind speed. To test the utility of pure IBA, another test is to use only the influence levels as observations. This will show how much the raw wind direction and speed contribute to learning.

7.3. Results

The experiment focused on evaluating the effectiveness of influence-based state abstraction in the context of AWC using a hexagonal wind farm layout under a dynamic wind process. Results were analyzed by comparing the REINFORCE algorithm both with and without IBA enhancements against a heuristic control strategy.

The following figure presented illustrates the energy output across different training steps for two main configurations: the standard multi agent REINFORCE algorithm and the REINFORCE algorithm enhanced with Influence-Based Abstraction (REINFORCE IBA). The IBA-enhanced version demonstrates a distinct advantage over the standard approach.

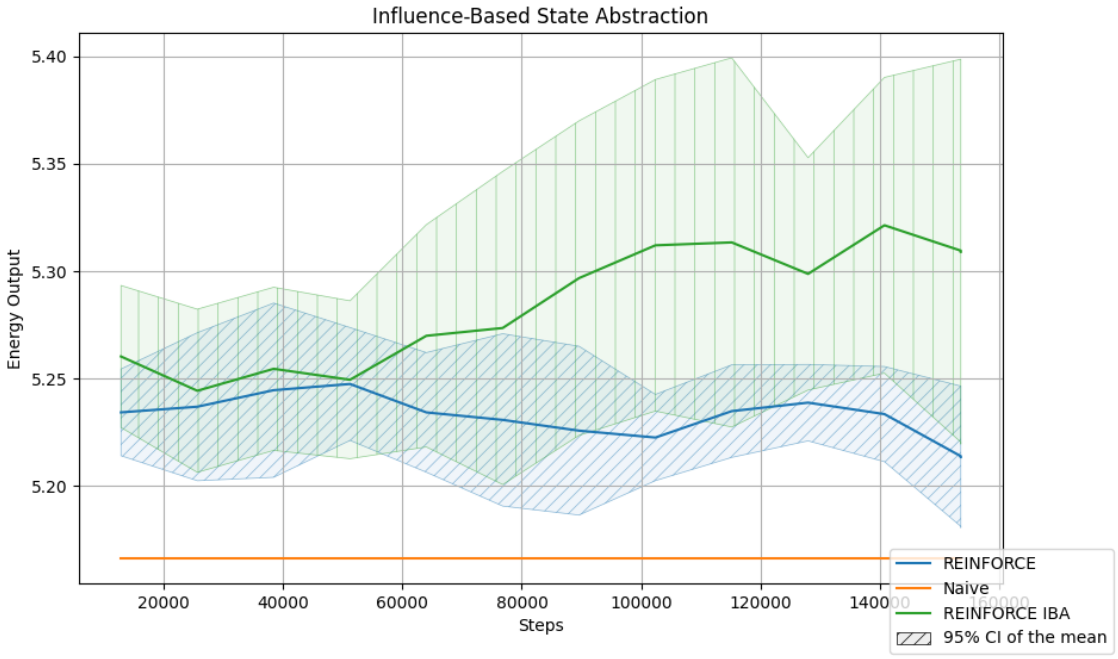

Figure 7.1: Energy Production per evaluation episode

Standard REINFORCE: This approach showed a relatively flat line in energy output throughout the experiment. It failed to adapt to the complexities of the dynamic wind environment and the wake interactions inherent in the hexagonal layout. Since it did outperform the naive strategy of facing the wind, it most likely converged to a suboptimal policy before the first evaluation The confidence interval remained narrow, indicating low variability but also suggesting that the learning did not progress effectively.

REINFORCE with IBA: In contrast, the IBA-enhanced REINFORCE algorithm shows a significant improvement in energy output over time. Initially, both methods performed similarly, but as learning progressed, REINFORCE IBA began to outperform the standard method significantly. The wider confidence interval in the IBA-enhanced approach indicates higher variability, which suggests that the algorithm was actively exploring and adapting to new strategies to improve performance.

The REINFORCE with IBA showed improved learning dynamics as evidenced by its upward trending energy output. This enhancement can be attributed to the IBA's ability to effectively show the influence relations within the wind farm, providing a more clear observation space that includes the abstracted influence levels of each turbine. By simplifying the complex interactions into manageable models, IBA allows REINFORCE algorithm to focus on the most impactful aspects of the environment, thus facilitating more effective learning and optimization of turbine settings.

The influence of adding IBA to the observation space increased the performance but did result in more instability. While the standard REINFORCE algorithm remained stable but underperforming, the

REINFORCE IBA reached higher performance levels but also had a much larger confidence interval. By examining the individual experiments, it is apparent that this is caused by most test cases learning an improved policy based on IBA, and performing near the top of the confidence interval, while others failed to learn at all and performed near the bottom. This means the mean shown in the graph is not representative of what one should expect when running this experiment.

These results have important implications for the application of AWC strategies in real-world settings. The ability of IBA to enhance learning by abstracting critical influence relationships can be particularly beneficial in optimizing the operation of wind farms where wake effects are a major concern. However, The experiment underscores that these techniques can be very unstable, this means that additional research is needed to determine better methods for learning roles.

7.4. Conclusion

In summary, the experiment demonstrates the value of integrating influence-based state abstraction into reinforcement learning frameworks for active wake control. By effectively simplifying and highlighting critical interactions within the wind farm, IBA facilitates simpler optimisation in complex multi-agent environments, leading to enhanced performance, but suffers from lack of stability.

In this chapter, we explored the application of IBA in optimizing AWC strategies within a wind farm environment. By abstracting critical influence relationships between turbines, IBA aims to enhance learning efficiency by focusing on the most impactful aspects of environmental interactions. This technique simplifies complex decision spaces, allowing the reinforcement learning algorithm to better adapt to dynamic wind conditions and the complex wake interactions characteristic of a hexagonal turbine layout.

Our findings indicate that while influence-based state abstraction leads to a more efficient learning process, it introduces a notable degree of instability in performance outcomes. The REINFORCE algorithm enhanced with IBA demonstrated a significant improvement in energy output over the baseline REINFORCE approach. However, the variability in performance, as evidenced by a wider confidence interval, suggests that the abstraction method, though powerful, does not consistently lead to stable learning dynamics. This instability is highlighted by the fluctuation in performance across different training instances, where some instances achieve near-optimal results and others fail significantly.

These results underline the potential trade-offs involved in employing state abstraction techniques like IBA in real-world applications. While the method can significantly increase the performance of learning algorithms by focusing on relevant environmental factors, the variability in outcomes can pose risks, especially in scenarios where consistent performance is critical. Moving forward, exploring alternative methods or refining IBA to enhance its stability could prove crucial for its practical deployment in managing complex systems like wind farms, where operational consistency is as important as optimization.

8

Conclusion

This thesis explored the potential of using Multi-Agent Reinforcement Learning to improve energy production in wind farms through Active Wake Control. AWC involves adjusting the yaw angles of upstream turbines to redirect their wakes, thus reducing the negative impact on downstream turbines. These losses can be large, and given the environmental end economic incentives to increase energy production, it follows that new optimisation techniques such as RL should be applied to this problem

Reinforcement Learning (RL) is presented as a method where agents learn optimal policies through interaction with the environment. This approach is suitable for AWC as it can adapt to the dynamic and complex nature of wind farms, as well as changing atmospheric conditions. The study specifically looks at MARL, where each turbine is modeled as an individual agent learning to optimize its performance while considering its impact on other turbines.

The primary research question is how influence information and domain knowledge can be used with MARL to increase wind farm energy production through AWC. Sub-questions focus on designing reward functions that encourage cooperation, using training regimes to stabilize performance, assigning task-specific roles to agents, and applying state-based abstractions to simplify learning. All of these have the aim of simplifying the optimisation problem, or highlighting important information to assist the agents' learning process.

The methodology involves using FLORIS to create a simulator for RL. We use various layouts to analyse different properties of wake effects, as well as the fitness of our solutions. We also various wind processes to take steps towards a fully dynamic wind process, and realistic simulation. Using this, we integrated new solutions into the standard policy gradient REINFORCE algorithm, to examine whether the interventions proposed lead to improved performance, and more optimal policies.

The experimental results demonstrate that difference rewards significantly enhance the performance of MARL by providing a more targeted approach to credit assignment among agents. Techniques like DrREINFORCE and Influence Rewards showed faster convergence and more consistent performance compared to standard REINFORCE. Staged training methods, where agents are trained sequentially or in groups based on their influence, also contributed to improved learning stability and efficiency. In the scenario where wind direction is dynamic, focusing on learning roles rather than complete policies was shown to improve performance. Furthermore, our influence based state abstraction experiment demonstrated that roles do not have to be provided explicitly. Roles can be approximated by the agent itself using state abstraction, but this does lead to increased volatility, and slightly decreased performance.

8.1. Conclusion

The study presented in this paper offers an examination of MARL with influence based modifications for AWC in wind farms. The study used difference rewards, training regimes, influence-based state abstraction, and role-based reinforcement learning to improve energy output and stabilize learning. The

research has showcased substantial improvements in both total energy output and learning stability.

The findings revealed that difference rewards solve the credit assignment problem for AWC, since the reward function is fully decomposable. This means agents know exactly which actions contributed to improved performance, and which hurt the farm performance. This did come at the cost of extra computational resources, however, the this can be mitigated through using influence rewards, which can be parallelized. Overall, difference rewards can be used to solve the credit assignment problem for any for any wind farm, and could potentially be combined with role-based reinforcement learning to optimize policies in more dynamic environments.

Staging training to stabilize the learning environment for agents did not lead to significant improvements in performance. Furthermore, training agents individually lead to worse performance, potentially due to fewer parameter updates. Alternating training between the individual agent and the whole ensemble did lead to a slightly narrower confidence interval, suggesting improved stability. The most important element of training regimes turned out to be the training order. It was showed that training the more influential upwind turbines first lead to a more stable environment for the downwind turbines, which improved stability. This may be harder to apply to more complex environments, but knowledge that training upwind turbines first could be combined with role based reinforcement learning to further improve stability.

Since difference rewards lead to an effective solution for AWC when the wind direction is constant, the second wind process with changing wind directions was introduced. To mitigate the increased complexity, role-based reinforcement learning was introduced, and was has proven to be effective in this context by leveraging domain knowledge to assign roles to agents, which simplifies learning. Assigning distinct roles based on each turbine's influence on downstream turbines led to a simplified approach to learning, which facilitated improvements in efficiency and stability. Furthermore, the more precisely the roles describe the task the higher the degree of specialisation, which leads to better policies. In addition, using domain knowledge for certain roles further simplifies learning, leading to improved stability.

Finally, since explicitly creating roles is not possible for a larger wind farm, we examined whether influence based state abstraction could lead to learning roles. Our findings revealed that influence-based abstraction enhanced the learning process by prioritizing and simplifying the interaction dynamics among agents. This method was demonstrated to improve learning outcomes significantly, as shown by the increased energy output compared to a standard REINFORCE algorithm. The ability of IBA to abstract influence relations within observations allows RL models to concentrate on the most impactful elements, resulting in more effective and efficient learning strategies. While it did assist in learning, by making roles explicit in the observations, it does not appear stable or scalable enough for larger farms, since the performance were too unstable.

In conclusion, the application of these advanced MARL strategies in wind farm management indicates a robust framework for enhancing energy output and operational efficiency. These methodologies allow for a nuanced control over the complex interactions between turbines, improving the adaptability and effectiveness of wake control strategies. Of the methods used in this paper, role-based reinforcement learning stands out as the most promising. While state abstraction did lead to improved learning, other methods should be investigated for learning roles. Combined with a method for identifying roles, which would be impossible to do manually at a large scale, this could lead to stable policies, which are scalable to larger wind farms.

8.2. Further Work

Looking ahead, several promising research directions could further the development of MARL applications in wind farm management:

Implementation of hypernetworks for role-based RL: Hypernetworks have been used in previous research to output netwrok parameters based on roles. By splitting AWC into learning roles based in observations, and learning network parameters based on roles, role-based reinforcement learning can be applied to larger wind farms.

Combination of intervention strategies: investigating the combined effects of different MARL strategies may uncover new synergies that could lead to improved performance. For instance, integrating

Influence Based Abstraction with role-based learning could refine the system's response to fluctuating environmental conditions, potentially leading to more robust control strategies.

Integration of turbine strain in reward functions: including the mechanical strain and durability of turbines in the reward calculations could ensure more sustainable and cost-effective operational strategies. This approach would help balance the immediate benefits of energy production with the long-term health and maintenance requirements of the turbines.

Utilization of high fidelity simulations: employing high fidelity simulations, such as Large Eddy Simulations, could enhance the accuracy of predictive models used in training and validating MARL strategies. These simulations can provide deeper insights into the fluid dynamics involved in wake interactions, which are crucial for developing more effective AWC techniques.

Scalability studies on larger wind farms: expanding the application of developed MARL strategies to larger and more diverse wind farms would test the scalability and generalizability of these approaches. Such studies would be vital in assessing the feasibility of widespread adoption of MARL techniques across the renewable energy sector.

Implementation of complex MARL algorithms: future studies could explore the use of more sophisticated MARL algorithms such as Multi-Agent Deep Deterministic Policy Gradient, and Multi-Agent Proximal Policy Optimization. These advanced algorithms might provide superior results in terms of agent coordination and efficiency, especially in handling the intricate dynamics of large-scale wind farms.

By advancing these key areas, subsequent research can deepen the understanding and capabilities of MARL, driving forward the optimization of wind farms globally. This would not only foster advancements in renewable energy technologies but also support the broader goal of enhancing the sustainability and efficiency of energy resources worldwide.

References

[1] US Energy Information Administration. *Where Wind Power is Harnessed*. 2023. URL: https://www.eia.gov/energyexplained/wind/where-wind-power-is-harnessed.php (visited on 05/17/2024).

[2] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press, 2024. URL: https://www.marl-book.com.

[3] Vyacheslav Alipov et al. *Towards Practical Credit Assignment for Deep Reinforcement Learning*. 2022. arXiv: 2106.04499 [cs.LG].

[4] Enrico GA Antonini, David A Romero, and Cristina H Amon. "Optimal design of wind farms in complex terrains using computational fluid dynamics and adjoint methods". In: *Applied Energy* 261 (2020), p. 114426.

[5] C. L. Archer, S. Mirzaeisefat, and S. Lee. "Quantifying the sensitivity of wind farm performance to array layout options using large-eddy simulation". In: *Geophysical Research Letters* 40 (18 2013), pp. 4963–4970. DOI: 10.1002/grl.50911.

[6] Aijun Bai, Siddharth Srivastava, and Stuart J. Russell. "Markovian State and Action Abstractions for MDPs via Hierarchical MCTS". In: *International Joint Conference on Artificial Intelligence*. 2016. URL: https://api.semanticscholar.org/CorpusID:15014895.

[7] Rebecca Jane Barthelmie et al. "Modelling the impact of wakes on power output in large offshore wind farms". English. In: *Extended Abstracts*. Ed. by Antonio Crespo, Gunner Chr. Larsen, and Emilio Migoya. EUROMECH Colloquium 508 on Wind Turbine Wakes ; Conference date: 20-10-2009 Through 22-10-2009. Universidad Politécnica de Madrid, 2009, pp. 84–85. ISBN: 978-84-7484-220-3.

[8] Richard Bellman. "A Markovian Decision Process". In: *Indiana Univ. Math. J.* 6 (4 1957), pp. 679–684. ISSN: 0022-2518.

[9] Ondrej Biza and Robert Platt. *Online Abstraction with MDP Homomorphisms for Deep Learning*. 2019. arXiv: 1811.12929 [cs.LG].

[10] Jacopo Castellini et al. "Difference rewards policy gradients". In: (2022). ISSN: 1433-3058. DOI: 10.1007/s00521-022-07960-5. URL: https://doi.org/10.1007/s00521-022-07960-5.

[11] Miguel Suau de Castro et al. "Influence-Based Abstraction in Deep Reinforcement Learning". In: 2019. URL: https://api.semanticscholar.org/CorpusID:198943538.

[12] M. Colby, W. Curran, and K. Tumer. "Approximating Difference Evaluations with Local Information (Extended Abstract)". In: *Proceedings of the Fourteenth International Joint Conference on Autonomous Agents and Multiagent Systems*. Istanbul, Turkey, May 2015.

[13] M. Colby et al. "Approximating Difference Evaluations with Local Knowledge (Extended Abstract)". In: *Proceedings of the Thirteenth International Joint Conference on Autonomous Agents and Multiagent Systems*. Paris, France, May 2014.

[14] P. Fleming et al. "Field test of wake steering at an offshore wind farm". In: *Wind Energy Science* 2.1 (2017), pp. 229–239. DOI: 10.5194/wes-2-229-2017. URL: https://wes.copernicus.org/articles/2/229/2017/.

[15] P. M. O. Gebraad et al. "Wind plant power optimization through yaw control using a parametric model for wake effects—a CFD simulation study". In: *Wind Energy* 19.1 (2016), pp. 95–114. DOI: https://doi.org/10.1002/we.1822. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/we.1822. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/we.1822.

[16] M. F. Howland, S. K. Lele, and J. O. Dabiri. "Wind farm power optimization through wake steering". In: *Proceedings of the National Academy of Sciences* 116 (29 2019), pp. 14495–14500. DOI: 10.1073/pnas.1903680116.

[17] Michael Howland, Sanjiva Lele, and John Dabiri. "Wind farm power optimization through wake steering". In: *Proceedings of the National Academy of Sciences* 116 (July 2019), p. 201903680. DOI: 10.1073/pnas.1903680116.

[18] Ember Energy Institute. "Statistical Review of World Energy". In: (2023). URL: https://www.energyinst.org/statistical-review.

[19] Jeewon Jeon et al. "MASER: Multi-Agent Reinforcement Learning with Subgoals Generated from Experience Replay Buffer". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 10041–10052. URL: https://proceedings.mlr.press/v162/jeon22a.html.

[20] Nan Jiang, Alex Kulesza, and Satinder Singh. "Abstraction selection in model-based reinforcement learning". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 2015, pp. 179–188.

[21] J Jonkman et al. "Definition of a 5-MW Reference Wind Turbine for Offshore System Development". In: (Feb. 2009). DOI: 10.2172/947422. URL: https://www.osti.gov/biblio/947422.

[22] Bangalore Ravi Kiran et al. "Deep Reinforcement Learning for Autonomous Driving: A Survey". In: *CoRR* abs/2002.00444 (2020). arXiv: 2002.00444. URL: https://arxiv.org/abs/2002.00444.

[23] Chenghao Li et al. "Celebrating Diversity in Shared Multi-Agent Reinforcement Learning". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 3991–4002. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/20aee3a5f4643755a79ee5f6a73050ac-Paper.pdf.

[24] Michael L. Littman. "Markov games as a framework for multi-agent reinforcement learning". In: *Machine Learning Proceedings 1994*. Ed. by William W. Cohen and Haym Hirsh. San Francisco (CA): Morgan Kaufmann, 1994, pp. 157–163. ISBN: 978-1-55860-335-6. DOI: https://doi.org/10.1016/B978-1-55860-335-6.50027-1. URL: https://www.sciencedirect.com/science/article/pii/B9781558603356500271.

[25] Ryan Lowe et al. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In: *CoRR* abs/1706.02275 (2017). arXiv: 1706.02275. URL: http://arxiv.org/abs/1706.02275.

[26] Marvin Minsky. "Steps toward Artificial Intelligence". In: *Proceedings of the IRE* 49.1 (1961), pp. 8–30. DOI: 10.1109/JRPROC.1961.287775.

[27] Seyed Iman Mirzadeh et al. "Understanding the role of training regimes in continual learning". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[28] Volodymyr Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[29] Ryan Nash, Reza Nouri, and Ahmad Vasel-Be-Hagh. "Wind turbine wake control strategies: A review and concept proposal". In: *Energy Conversion and Management* 245 (2021), p. 114581. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2021.114581. URL: https://www.sciencedirect.com/science/article/pii/S0196890421007573.

[30] Ryan Nash, Reza Nouri, and Ahmad Vasel-Be-Hagh. "Wind turbine wake control strategies: A review and concept proposal". In: *Energy Conversion and Management* 245 (2021), p. 114581. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2021.114581. URL: https://www.sciencedirect.com/science/article/pii/S0196890421007573.

[31] Grigory Neustroev et al. "Deep Reinforcement Learning for Active Wake Control". In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '22. Virtual Event, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems, 2022, pp. 944–953. ISBN: 9781450392136.

[32] Frans Oliehoek, Stefan Witwicki, and Leslie Kaelbling. "Influence-Based Abstraction for Multiagent Systems". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 26.1 (Sept. 2021), pp. 1422–1428. DOI: 10.1609/aaai.v26i1.8253. URL: https://ojs.aaai.org/index.php/AAAI/article/view/8253.

[33] Liviu Panait and Sean Luke. "Cooperative Multi-Agent Learning: The State of the Art". In: *Autonomous Agents and Multi-Agent Systems* 11 (2005), pp. 387–434. URL: https://api.semanticscholar.org/CorpusID:19706.

[34] Shubham Pateria et al. "Hierarchical Reinforcement Learning: A Comprehensive Survey". In: *ACM Comput. Surv.* 54.5 (June 2021). ISSN: 0360-0300. DOI: 10.1145/3453160. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3453160.

[35] Rafael Pina et al. "Staged Reinforcement Learning for Complex Tasks Through Decomposed Environments". In: *Intelligent Systems and Pattern Recognition*. Ed. by Akram Bennour, Ahmed Bouridane, and Lotfi Chaari. Cham: Springer Nature Switzerland, 2024, pp. 141–154. ISBN: 978-3-031-46338-9.

[36] Scott Proper and Kagan Tumer. "Modeling difference rewards for multiagent learning". In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*. AAMAS '12. Valencia, Spain: International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 1397–1398. ISBN: 0981738133.

[37] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall, 2010.

[38] R. Saidur et al. "Environmental impact of wind energy". In: *Renewable and Sustainable Energy Reviews* 15.5 (2011), pp. 2423–2430. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2011.02.024. URL: https://www.sciencedirect.com/science/article/pii/S1364032111000669.

[39] J Schreiber et al. "Verification and Calibration of a Reduced Order Wind Farm Model by Wind Tunnel Experiments". In: *Journal of Physics: Conference Series* 854.1 (May 2017), p. 012041. DOI: 10.1088/1742-6596/854/1/012041. URL: https://dx.doi.org/10.1088/1742-6596/854/1/012041.

[40] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. "Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving". In: *CoRR* abs/1610.03295 (2016). arXiv: 1610.03295. URL: http://arxiv.org/abs/1610.03295.

[41] Adam Sigal, Hsiu-Chin Lin, and AJung Moon. *Improving Generalization in Reinforcement Learning Training Regimes for Social Robot Navigation*. 2024. arXiv: 2308.14947 [cs.RO]. URL: https://arxiv.org/abs/2308.14947.

[42] Rolf A. N. Starre, Marco Loog, and Frans A. Oliehoek. "Model-Based Reinforcement Learning with State Abstraction: A Survey". In: *Artificial Intelligence and Machine Learning*. Ed. by Toon Calders et al. Cham: Springer Nature Switzerland, 2023, pp. 133–148. ISBN: 978-3-031-39144-6.

[43] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

[44] G. Tesauro and J. O. Kephart. "Pricing in Agent Economies Using Multi-Agent Q-Learning". In: *Autonomous Agents and Multi-Agent Systems* 5 (3 2002), pp. 289–304. DOI: 10.1023/a:1015504423309.

[45] Jared J Thomas et al. "Comparison of wind farm layout optimization results using a simple wake model and gradient-based optimization to large eddy simulations". In: *AIAA Scitech 2019 Forum*. 2019, p. 0538.

[46] M Vali et al. "Large-eddy simulation study of wind farm active power control with a coordinated load distribution". In: *Journal of Physics: Conference Series* 1037.3 (June 2018), p. 032018. DOI: 10.1088/1742-6596/1037/3/032018. URL: https://dx.doi.org/10.1088/1742-6596/1037/3/032018.

[47] Deepak A. Vidhate and Parag Kulkarni. "Cooperative multi-agent reinforcement learning models (CMRLM) for intelligent traffic control". In: *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*. 2017, pp. 325–331. DOI: 10.1109/ICISIM.2017.8122193.

[48] Tonghan Wang et al. "RODE: Learning Roles to Decompose Multi-Agent Tasks". In: *CoRR* abs/2010.01523 (2020). arXiv: 2010.01523. URL: https://arxiv.org/abs/2010.01523.

[49] Yihan Wang et al. "DOP: Off-Policy Multi-Agent Decomposed Policy Gradients". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=6FqKiVAdI3Y.

[50] Dezhi Wei et al. “Parametric study of the effectiveness of active yaw control based on large eddy simulation”. In: *Ocean Engineering* 271 (2023), p. 113751. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2023.113751. URL: https://www.sciencedirect.com/science/article/pii/S002980182300135X.

[51] George Weiss. “Dynamic Programming and Markov Processes. Ronald A. Howard. Technology Press and Wiley, New York, 1960. viii + 136 pp. Illus. $5.75.” In: *Science* 132.3428 (1960), pp. 667–667. DOI: 10.1126/science.132.3428.667.a. eprint: https://www.science.org/doi/pdf/10.1126/science.132.3428.667.a. URL: https://www.science.org/doi/abs/10.1126/science.132.3428.667.a.

[52] R. J. Williams. “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In: *Machine Learning* 8 (3-4 1992), pp. 229–256. DOI: 10.1007/bf00992696.