Tsinghua Science and Technology
Doi: xxxx

Research Article

Adversarial Attack and Defense in Multi-Agent Deep Reinforcement Learning for Active Distribution Network Dispatching

Zhuocen Dai, Mao Tan (✉), Yi Su, Xiao Liu, Yin Yang (✉) and Kang Li
(Please place (✉) after the corresponding author; use full names, not initials, for all the authors' names)

Abstract Multi-agent Deep Reinforcement Learning-based (MADRL) optimal dispatching for active distribution networks (ADNs) represents a future trend, necessitating robust security measures due to the sensitivity of MADRL. However, limited research addresses this issue. This paper introduces Multi-Agent Directed Shift Attack (MADSA), a powerful attack strategy for MADRL in ADNs, which targets single and full agents. The single-agent attack maximizes the deviation of a single agent's strategy before and after the attack, leading to ADN degradation. And the full-agent attack unifies the action direction of all agents to achieve maximum degradation. To counter MADSA, we propose an adversarial defense method named Multi-Agent Gradient Leveling Defense (MAGLD) with Gradient Leveling Regularization, which enhances the robustness of the defense strategy. Case studies show MADSA can degrade the ADN under continuous and single-step attacks, with even the small attack amplitudes significantly increasing power losses and causing overvoltage, surpassing existing methods, such as fast gradient sign method and noise attacks. The proposed defense strategy effectively mitigates these attacks, offering forward-looking considerations for the security of artificial intelligence based control in ADNs.

Keywords Multi-agent Deep Reinforcement Learning, Active Distribution System, Optimal Dispatching, Adversarial Attack.

• Zhuocen Dai is with Hunan Key Laboratory for Computation and Simulation in Science and Engineering, National Center for Applied Mathematics in Hunan, Xiangtan University, Xiangtan 411105, Hunan, China. E-mail: zhuocend@protonmail.ch.
• Yin Yang is with Hunan International Scientific and Technological Innovation Cooperation Base of Computational Science, Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education, Xiangtan University, Xiangtan 411105, Hunan, China. E-mail: yangyinxtu@xtu.edu.cn.
• Mao Tan, Xiao Liu and Yi Su are with the School of Automation and Electronic Information, Xiangtan University, Xiangtan 411105, China. E-mail: mr.tanmao@gmail.com; liuxiao730@outlook.com; suyi2018@xtu.edu.cn.
• Kang Li is with the School of Electronic and Electrical Engineering, University of Leeds, LS2 9JT, Leeds, UK E-mail: k.li1@leeds.ac.uk.
* To whom correspondence should be addressed.
Manuscript received: 2026-02-27; revised: 2026-03-16 and 2026-04-14; accepted: 2026-05-12

Nomenclature

Indexes

i	Index of agent.
j	Index of attacked agent.
l, m	Index of node.
k	Index of iteration in MADSA.
c	Index of disturbance in candidate set.
h	Index of dimensions.

Sets

$\mathcal{A}$	Set of the agents.
$\mathcal{A}_t^{att}$	Set of attacked agents at time t.
$\mathcal{O}_i$	Candidate set of disturbance $\boldsymbol{\delta}_{i,t}$.

Parameters

ϵ	Threshold for disturbance between the FDI and the true value, making the FDI hard to be detected.
a_i^{cmax}	Maximum limits of charge active power of ESS.
a_i^{dmax}	Maximum limits of discharge active power of ESS.
n	Number of the agents.
ϕ	Maximum deviation limit of SOC from the initial value during operation
ρ	Penalty coefficient when the SOC is not satisfied.
β	Voltage deviation threshold in ADNs.
t^{att}	Execution time slot of the attack
H	Number of nodes in ADN.
K	Number of iterations in MADSA.
Z	Size of candidate set $\mathcal{O}_i$.
λ	Regularization coefficient in defense.

Variables

π_i	Denoting the policy of agent i.
$\mu_{\pi_i}(\boldsymbol{o}_{i,t})$	Output of the forward propagation function π_i when the input is $\boldsymbol{o}_{i,t}$.
$r_{i,t}$	Reward of agent i at time t.
$r_i(\boldsymbol{o}_{i,t}, a_{i,t})$	Reward function of agent i.
$u_{j,t}$	Voltage (p.u.) of node j at time t.
$p(a_{i,t})$	Penalty function of the SOC constraint of ESSs.
$SOC_{i,t}$	State of charge of the i-th ESS at time t.
$\boldsymbol{P}_t^{PV}$	Active power output vector of PVs at time t.
$\boldsymbol{P}_t^{WT}$	Active power output vector of WTs at time t.

$\boldsymbol{P}_t^{LD}$	Active power demand vector of loads at time t.
$\boldsymbol{\delta}_{i,t}$	The disturbance vector of attack samples.
$\mathcal{L}_{i,t}^{att}$	Loss function of adversarial attacks.
θ_i, θ	Parameters of agent i and parameters of agents.
$\mathcal{L}(\theta)$	Loss function of MADRL in defense training.
$\mathcal{L}_i(\theta_i)$	Loss function of agent i in defense training.
$\mathcal{L}_{smooth}(\theta)$	Smoothed loss function by introducing GLR.
$\boldsymbol{s}_t$	The vector of state at time t.
$\boldsymbol{o}_t$	The vector of observation at time t.
$\boldsymbol{a}_t$	The vector of actions at time t.

1 Intoduction

Deep Reinforcement Learning (DRL) optimizes strategy through real-time interaction with the environment [1]. It can make decisions in complex systems [2] without relying on predictions of future states [3]. DRL has been widely used in systems requiring real-time complex decisions, such as AlphaGo [4], robotic autonomous navigation and planning [5], and intelligent traffic management [6]. In the field of active distribution networks (ADNs), researchers employ DRL to address various challenges including optimal scheduling [7], dynamic reconstruction [8], post-disaster recovery [9], and voltage control [10]. The significance of optimal dispatching lies in its ability to maintain voltage security and operational economy amidst the increasing integration of volatile renewable energy sources. A DRL-based dispatching method significantly enhances ability to handle complex issues and is expected to see further promotion and application in the future.

Although DRL has significant advantages in optimization decision-making, it is highly sensitive to the environment, making it challenging to establish a stable optimization in dynamic environment [11]. Consequently, when the physical topology of the ADN changes, which is common in the real world, single-agent DRL requires retraining, complicating its application. To address this issue, many scholars have proposed multi-agent DRL, which is more robust than the single-agent approach and can continue to operate despite local damage, such as communication loss or physical failure of some agents [12]. Moreover, multi-agent DRL supports plug-and-play functionality, meaning that the removal or addition of agents does not affect the trained agents [13]. Therefore, multi-agent DRL has been promoted in ADNs [14–16].

Given the potential application of multi-agent Deep Reinforcement Learning-based (MADRL) optimal dispatching model, its related security considerations cannot be ignored, despite the existence of monitoring methods [17]. However, there are few studies on the attack and defense of DRL-based ADNs. Table 1 provides a summary of typical attacks on DRL-based model of power grid. As seen in the table, references [18], [19], [20] and [21] discuss large-scale False Data Injection (FDI), Fast Gradient Sign Method (FGSM), and noise injection on ADNs based on single-agent DRL, achieving certain levels of attack effectiveness. However, large-scale FDI can be easily detected and intercepted, so it lacks engineering feasibility, while the attack intensity of gradient perturbation and noise injection is not high. References [14] and [22] address attacks on MADRL ADNs, focusing on the impact of communication congestion/ failures states and denial-of-service (Dos) attacks on multi-agent model. In summary, there is no existing research on attacks targeting MADRL optimal dispatching of ADNs which involves implementing hard-to-detect FDI that can cause significant response changes in the ADNs.

The typical defense method against adversarial attacks involves incorporating adversarial samples into the training data for retraining, which helps develop a more robust model. However, this approach can make convergence more challenging and decrease the accuracy [23]. References [18], [19] demonstrated the generation of adversarial samples and their defense effects in power grid optimization control using single-agent approaches. References [24] also introduces multiscale generative adversarial networks for fault diagnosis. However, due to the inherently weaker convergence of multi-agent models compared to single-agent ones [25], simply adding adversarial samples for defense in ADNs proves difficult to implement in practice [26]. Therefore, it is crucial to develop a multi-agent defense framework that minimizes performance degradation while effectively countering strong external network attacks in a timely and accurate manner.

In general, MADRL optimal dispatching of ADNs, as a future trend, lacks sufficient security research. Specifically, current research on attacks and defenses for DRL-based optimal dispatching in ADNs is quite limited and primarily focuses on basic FGSM and noise-based methods. There are even fewer studies on MADRL dispatching of ADNs, despite its greater instability and need for more targeted defense strategies. Retraining the model by incorporating adversarial samples is one strategy to improve DRL defense. However, it is necessary to theoretically prove the feasibility of MADRL defense after incorporating adversarial samples and to address the increased difficulty of MADRL training caused by these samples. Therefore, this paper proposes a powerful attack algorithm called MADSA and a corresponding defense strategy named MAGLD, and provides theoretical proof of their effectiveness. The contributions are as follows:

1) To the best of our knowledge, this is the first study

Table 1 Selected References Review of DRL-Based ADN Dispatching and its attack and defense in related fields

Ref	Number of agents		Attack Category		Adversarial Attack method			Defense	Tasks
	Single	Multi	FDI	Dos	Noise	FGSM	MADSA		
[7]	✓								Dispatching of ADNs
[8]	✓								Dynamic Reconstruction
[9]	✓								Post-disaster recovery
[10]	✓								Voltage control
[15]		✓							Voltage regulation
[16]		✓							Energy management
[18]	✓		✓		✓			✓	Inverter optimization
[19]	✓		✓			✓			Optimal power flow
[21]	✓		✓		✓				Topology optimization
[14]		✓		✓					Dispatching of Energy storage systems
[22]		✓		✓				✓	Frequency control based on inverter
This paper		✓	✓				✓	✓	**Dispatching of ADNs**

to investigate FDI attacks on MADRL-based power system dispatching. It introduces subtle false data to perform both continuous and single-step attacks on single and multiple agents, maximizing the impact on agents' actions and causing operational degradation in ADNs.

2) A Multi-Agent Directed Shift Attack (MADSA) strategy is proposed. This strategy can target a single agent to maximize action deviation, or conduct a full-agent attack by directing all agents' actions to shift in the same direction, thereby maximizing voltage deviations in the ADNs.

3) A Multi-Agent Gradient Leveling Defense (MAGLD) strategy is proposed to counter MADSA. To enhance adversarial robustness, a gradient leveling regularization-based adversarial training method is introduced, and its feasibility is theoretically proven.

The remainder of this paper is organized as follows. Section 2 introduces the MADSA method targeting MADRL optimal dispatching of ADNs. In Section 3, the defense strategy named MAGLD is proposed to minimize performance degradation while effectively countering the attacks. Section 4 presents the simulation results and discussions. Finally, the study is concluded in Section 5.

2 Attack Method on Multi-agent DRL-based ADNs Dispatching

The attack on the MADRL optimal dispatching of the ADNs involves injecting false data that is only slightly different from the true values. This manipulation induces the equipment to take inappropriate actions based on the agents' decisions from the FDIs [27], ultimately causing the ADN to deteriorate. Thus, this section begins by designing the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) for optimal dispatching of ADNs. Following this, the classification of FDI attacks based on attack timing and objects is introduced. Existing research offers a substantial number of theories and methods on adversarial attacks. In this section, a targeted analysis and derivation of adversarial attacks in MADRL based ADN dispatching is conducted. Finally, the proposed powerful attack method, called the Multi-Agent Directed Shift Attack (MADSA), is presented.

2.1 Dec-POMDP for optimal dispatching of ADNs

The optimal dispatching for ADNs encompasses both safety and economic dispatching, sharing the same constraints within the same topology but differing in objectives [28]. Considering the high penetration of Distributed Energy Sources (DESs) in ADNs, which can cause voltage deviations, a typical Dec-POMDP to minimize these deviations is introduced. In this model, each Energy Storage System (ESS) is controlled by a DRL agent, and every DES is fully absorbed. Dec-POMDP is a fundamental framework for multi-agent decision-making in distributed environments where agents have only partial observability. Its key components include: 1) The set of agents or policies. 2) state and observation - while the system has a global state, each agent receives only local observations; 3) action - each agent selects actions based on its partial observations, with joint actions collectively affecting the environment; 4) reward - typically a shared team reward signal guides cooperative behavior. These interacting elements enable Dec-POMDP to model both partial observability in real-world scenarios and the complexity of multi-agent coordination, making it an essential foundation for MADRL research. The details of the Dec-POMDP are as follows.

1) Agent Set $\mathcal{A}$: The set of all agents $\mathcal{A}$ in the system. Denoting the policy of agent i as π_i , the joint policy of all agents is represented as $\Pi = \pi_1 \times \pi_2 \times \cdots \times \pi_n$, where n is the total number of agents.

2) State $\boldsymbol{s}_t$ and Observation $\boldsymbol{o}_t$: The state is formulated as all the information of the system:

$$\boldsymbol{s}_t = \left[\boldsymbol{SOC}_t, \boldsymbol{P}_t^{PV}, \boldsymbol{P}_t^{WT}, \boldsymbol{P}_t^{LD}\right], \tag{1}$$

where $\boldsymbol{SOC}_t$ represents the state of charge (SOC) of the ESSs, $\boldsymbol{P}_t^{PV}, \boldsymbol{P}_t^{WT}, \boldsymbol{P}_t^{LD}$ represent the output active power of Photovoltaic (PV), Wind Turbine (WT), and the loads respectively at time *t*. Considering that the ESSs are controlled in a distributed manner (multi-agent), each agent can only observe its own operational state. The observation vector of each agent i can be represented as

$$\boldsymbol{o}_{i,t} = \left[SOC_{i,t}, \boldsymbol{P}_t^{PV}, \boldsymbol{P}_t^{WT}, \boldsymbol{P}_t^{LD}\right], \tag{2}$$

where $SOC_{i,t}$ is SOC of i-th ESS. In this paper, the state is also the union of the observations of all agents, represented as $\boldsymbol{s}_t = \bigcup_{i\in\mathcal{A}} \boldsymbol{o}_{i,t}$.

3) Action $\boldsymbol{a}_t$: Each agent i makes an action $P_{i,t}^{ES} = a_{i,t} = \mu_{\pi_i}(\boldsymbol{o}_{i,t})$ as the charging/discharging active power of ESS *i*. Considering the charging and discharging power limit of ESS, $a_{i,t} \in [-a_i^{dmax}, a_i^{cmax}]$, where a_i^{dmax} and a_i^{cmax} are the maximum limits of charge and discharge power. The action of the entire system is then defined as the joint action of all agents, represented as $\boldsymbol{a}_t = a_{1,t} \times \cdots \times a_{n,t}$.

4) Reward $r_{i,t} = r_i(\boldsymbol{o}_{i,t}, a_{i,t})$: In multi-agent DRL, the immediate reward for each agent should be equal to the global objective. Thus, this reward can be represented as Equation (3) shows in the optimal dispatching for ADNs.

$$r_i(\boldsymbol{o}_{i,t}, a_{i,t}) = p(SOC_{i,t}|a_{i,t}) + \sum_{j=1}^{H} f(u_{j,t}|\boldsymbol{a}_t),\ \forall\, i \in \mathcal{A}\ . \tag{3}$$

s.t.

$$p\left(SOC_{i,t}|a_{i,t}\right) = \begin{cases} 0 & |SOC_{i,t} - SOC_{i,0}| \leqslant \phi, \\ |SOC_{i,t} - SOC_{i,0}| \cdot \rho & \text{otherwise}, \end{cases} \tag{4}$$

$$f\left(u_{j,t}|\boldsymbol{a}_t\right) = \begin{cases} \frac{|u_j - 1|}{\beta} & 1-\beta \leqslant u_i \leqslant 1+\beta, \\ 1 & else, \end{cases} \tag{5}$$

where *H* is the number of nodes in ADNs. ϕ is the maximum deviation limit of SOC from the initial value during operation and ρ is the penalty coefficient. β is the voltage deviation threshold and $[1-\beta, 1+\beta]$ is the acceptable range of voltage. $u_{j,t}$ is the voltage (p.u.) of node *j* at time *t*.

The Equation (3) shows the reward of *i*-th agent at time *t*, Equation (4) shows the penalty function of the SOC constraint of ESSs and Equation (5) shows the dispatching objective, represented by the voltage deviation.

Thus, the objective dispatching function of ADNs can be formulated as $F^{obj} = \sum_{t=1}^{T}\sum_{j=1}^{H} f\left(u_{j,t}|\boldsymbol{a}_t\right)$.

After constructing the Dec-POMDP, various established methods can be employed to solve it, such as MADDPG [28], MAPPO [29], and Qmix [30]. It is important to note that the focus of this article is on attack and defense rather than the MADRL solution itself. Therefore, the solution part will not be reiterated and the MADDPG will be used for model training and solution. Additionally, to consider the generalization of the attack and defense, the paper verifies the effects of attack and defense on different full-agent solution methods in the Section IV.

2.2 Classification of FDI Attack

First, we consider the impact of attacks on agent observation. When a small perturbation $\boldsymbol{\delta}_{i,t}$ is injected to the observation $\boldsymbol{o}_{i,t}$, the change of output action by agent i can be expressed using Taylor expansion:

$$a_i = \mu_{\pi_i}(\boldsymbol{o}_{i,t} + \boldsymbol{\delta}_{i,t}) \approx \mu_{\pi_i}(\boldsymbol{o}_{i,t}) + \nabla_{\boldsymbol{o}_{i,t}}\mu_{\pi_i}(\boldsymbol{o}_{i,t}) \cdot \boldsymbol{\delta}_{i,t}, \tag{6}$$

where $\nabla_{\boldsymbol{o}_{i,t}}\mu_{\pi_i}(\boldsymbol{o}_{i,t})$ is the gradient of the policy function with respect to the input. A large gradient value means that the policy function is very sensitive to the input, and a small perturbation $\boldsymbol{\delta}_{i,t}$ may cause a drastic change in the action. This sensitivity, combined with the lack of gradient smoothness in vanilla MADRL, creates a significant vulnerability that attackers can exploit to induce systemic instability. However, in vanilla MADRL training, there is no guarantee that the gradient is smooth. This suggests that the agent is sensitive to the input observations.

Since the output of any agent in a multi-agent system affects the next system state, thereby affecting the observation and decision-making of other agents, the perturbation in MADRL will be propagated and amplified by the interaction between all agents, which can be represented as

$$\boldsymbol{s}_{t+W} \approx \boldsymbol{s}'_{t+W} + \sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{A}_t^{att}}\sum_{w\in[0,W-1]} \nabla_{\boldsymbol{o}_{i,t+w}}\mu_{\pi_{t+w}}(\boldsymbol{o}_{i,t}) \cdot \boldsymbol{\delta}_{j,t+w}, \tag{7}$$

where $\boldsymbol{s}_{t+W}$ and $\boldsymbol{s}'_{t+W}$ are the origin and attack state after W timesteps. In particular, if $|\mathcal{A}_t^{att}| > 1$ and the output disturbance is in the same direction, the accumulation of multiple disturbances will result in stronger overall effect of the disturbance and greater deviation. Therefore, the FDI

attacks in MADRL is introduced and classified based on the number of attacked agents and the number of time steps.

As shown in Figure 1, the FDI attacks on the multi-agent system discussed in this paper can be seen as tampering with the agents' observations. Notably, the altered observations $\boldsymbol{o}'_{i,t} = \boldsymbol{o}_{i,t} + \boldsymbol{\delta}_{i,t}$ may be only slightly different from the original observations $\boldsymbol{o}_{i,t}$, making them challenging to detect [31]. These tampered inputs can lead the agents to make decisions that deviate significantly from their original actions, resulting in an attacked action $\boldsymbol{a}'_t = a'_{1,t} \times a'_{2,t} \times \cdots \times a'_{n,t}$. If these compromised decisions (representing the action of ESSs in this paper) are implemented, they can cause degradation in the ADNs, manifesting as increased voltage deviations and power loss.

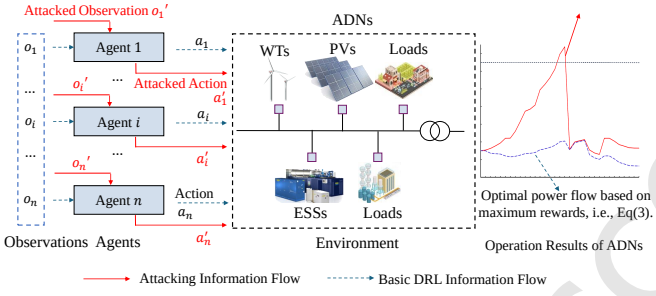


Fig. 1 Attack diagram for MADRL of ADNs.

The stealthiness of the adversarial perturbation $\boldsymbol{\epsilon}$ is defined by its ability to bypass the standard Bad Data Detection (BDD) mechanism. The BDD identifies anomalies using the weighted sum of squared residuals:

$$J(\hat{\mathbf{x}}) = \sum \left(\frac{z_i - h_i(\hat{\mathbf{x}})}{\sigma_i} \right)^2 \leqslant \tau \tag{8}$$

where z_i is the measurement, σ_i is the standard deviation of sensor noise, and τ is the detection threshold derived from the χ^2 distribution.

An attack is considered sufficiently stealthy if $J(\hat{\mathbf{x}}_\epsilon) \leqslant \tau$ is satisfied. Therefore, $\boldsymbol{\epsilon}$ should be constrained to the same order of magnitude as the measurement noise σ, the resulting residual shift remains largely indistinguishable from stochastic fluctuations, ensuring that the attack is operationally hard to detect by the monitor.

Before detailing the attack algorithm, it is essential to classify the FDI attacks according the attacking times and attacking objectives:

From the perspective of time dimension, the FDI attacks on the proposed MADRL can be defined as two types:

1) **Single-step Attack.** The attacker may inject malicious data only at one time slot. It should be noted that since DRL is the optimization of a decision trajectory, a single-step attack may also affect subsequent decisions. The objective of single-step attack can be formulated as

$$\max_{\Pi} \left[\mathcal{R}\left(\boldsymbol{s}'_{t^{att}}, \boldsymbol{a}'_{t^{att}}\right) + \sum_{t=t^{att}+1}^{T} \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t) \right], \tag{9}$$

where t^{att} is the execution time slot of the attack. $s'_{t^{att}}$ is the attacked state of the system and $\boldsymbol{a}'_{t^{att}} = \mu_\Pi(s'_{t^{att}})$ is the joint action based on $s'_{t^{att}}$. $\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is the MADRL risk function, representing the instability of agents consistently executing the attacked a'_i in time t. Specifically, the systemic risk function $\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is defined as

$$\mathcal{R}\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) = \mathbb{I}\,|u_{i,t} - 1| > \beta, \exists i \in H, \tag{10}$$

where $\mathbb{I}$ is a boolean indicator function that is $\mathcal{R}\left(\boldsymbol{s}_t, \boldsymbol{a}_t\right) = 0$ when all node voltages in the system are within the appropriate range, and equal to 1 otherwise.

2) **Continuous Attack.** Considering the extreme case that the attacker can attack in all time slots with slight FDI, the objective of continuous attack can be defined as

$$\max_{\Pi^{att}} = \left[\sum_{t=0}^{T} \mathcal{R}\left(\boldsymbol{s}'_t, \boldsymbol{a}'_t\right) \right]. \tag{11}$$

In this case, the attacker can accumulate deviations through multiple steps of attacks to make the entire decision trajectory violate the optimal power flow as much as possible.

From the number of attacked agents, the FDI attacks can be defined as two types:

1) **Full-Agent Attack.** The attacker injects malicious data into all agents at time t, which is defined as $\boldsymbol{s}'_t = \bigcup_{i \in \mathcal{A}} \boldsymbol{o}'_{i,t}$. $o'_{i,t}$ represents the attacked observation of agent i, which should satisfy $||o'_{i,t} - o_{i,t}||_2 < \epsilon$ to make it difficult to be detected. Please note that, in the context of our study, an attack on an agent is defined under the assumption that all dimensions of its input observation are susceptible to perturbation.

2) **Single-Agent Attack.** The attacker only injects malicious information into one agent at time t. Denoting the attacked agent as j, the system state is defined as $\boldsymbol{s}'_t = \left(\bigcup_{i \in (\mathcal{A}-j)} \boldsymbol{o}_{i,t}\right) \bigcup \boldsymbol{o}'_{j,t}$.

It is important to note that, in single-agent attack, since only specific agents are available to attack, there is no process of selecting which agent to attack. On the contrary, if the attacker could choose which agent to attack, they could perform FDI on all agents, which would lead them to execute a full-agent attack for a better attack result, rather than a single-agent attack. Therefore, there is no situation where the attacker can selectively attack specific agents. The single-agent attack discussed in the manuscript refers to how the attacker can make slight modifications to its observations in order to

maximize the voltage deviation in the system when it is known which agent can be attacked.

In addition, according to the information available to the attacker, attacks can be classified based on the attacker knowledge as follows:

1) **White-box Attack.** The attacker has full access to the system's internal information, including its architecture, parameters, and gradients. This allows the attacker to compute precise perturbations for maximum impact, representing the most severe threat scenario.

2) **Gray-box Attack.** The adversary has partial knowledge of the system, such as the model architecture or data distribution, but lacks access to the exact parameters.

3) **Black-box Attack.** The attacker can only interact with the system through input-output queries, treating it as an opaque box.

In the subsequent experimental validation, all attack methods are conducted under the white-box setting by default unless otherwise specified. In addition, black-box attack experiments are also provided to evaluate the effectiveness of the proposed method under limited knowledge conditions. It is worth noting that the gray-box setting lies between the white-box and black-box cases, with partial knowledge of the target system. Since its configuration is not uniquely defined, we do not explicitly conduct separate gray-box experiments. Instead, the white-box and black-box settings can be regarded as two representative extreme cases, and the performance under gray-box conditions is expected to fall between them.

2.3 Multi-Agent Directed Shift Attack method

This section details the MADSA method, which employs False Data Injection (FDI) to mislead the ADN dispatching system into sub-optimal operational decisions. The design of MADSA is guided by two strategic principles: individual agent impact and collective multi-agent disruption. From a single-agent perspective, the $\mathbf{MADSA}^{s}$ model identifies adversarial samples that drive a specific agent toward its worst-case decision. From a systemic perspective, the $\mathbf{MADSA}^{-}$ and $\mathbf{MADSA}^{+}$ models execute a coordinated directed attack, forcing the actions of all agents to shift simultaneously in the same direction. This synchronization maximizes the overall system voltage deviation as formulated in Eq. ((3)). The conceptual process of MADSA is illustrated in Figure 2.

At any time *t*, let the set of attacked agents be $\mathcal{A}_t^{att}$. For each agent $i \in \mathcal{A}_t^{att}$, we definite a loss function $\mathcal{L}_{i,t}^{att}$ for adversarial attack. In the context of a single-agent attack in MADRL, when the output of the agent after the attack

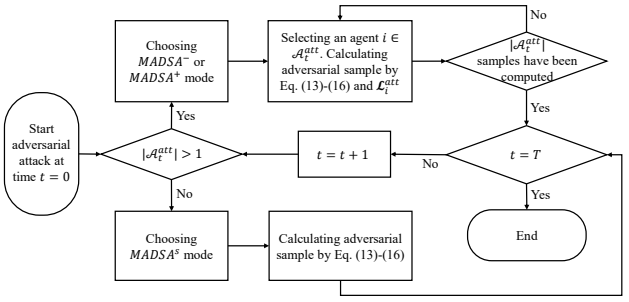


Fig. 2 Flow chart of MADSA.

$\boldsymbol{a}'_{i,t}$ deviates from the most from the original action $\boldsymbol{a}_{i,t}$, the decision of the agent is considered the worst. Then we can define the attack mode of single agent.

$\mathbf{MADSA}^{s}$ for single-agent attack. The attacker shifts the attacked action $a'_{i,t} = \mu_{\pi_i}(\boldsymbol{o}'_{i,t})$ as much as possible from the original action $a_{i,t}$ and finds the corresponding attack sample $\boldsymbol{o}'_{i,t}$. When the action vector of each agent $a_{i,t}$ is maximally distant from the original action $a_{i,t}$, the shift in the joint action also reaches its maximum. Therefore, for each agent $i \in \mathcal{A}_t^{att}$, the function is set as $\mathcal{L}_{i,t}^{att} = \max||a'_{i,t} - a_{i,t}||$.

However, in a multi-agent system, the maximum deviation of each single agent's action does not necessarily mean the system is in the worst state [32]. For example, in a power system with two agents at time t, if $a_{1,t} > 0$ (charge) and $a_{2,t} < 0$ (discharge), after attacking both agents according to MADSA, there could be $a'_{1,t} < 0$ (discharge) and $a'_{2,t} > 0$ (charge). Since charging and discharging have opposite effects, the overall system goal is not the worst.

Therefore, attacks on MADRL need to be designed with consideration of their actual physical impact.

The voltage relationship in radial distribution networks can be rigorously described using the linearized DistFlow model [29], which is a simplified power flow model derived for radial topologies under typical assumptions such as small voltage deviations and negligible line losses. For a branch connecting an upstream node l to a downstream node m with impedance $r_{lm} + jx_{lm}$, the voltage drop is approximated as $V_l - V_m \approx (r_{lm}P_{lm} + x_{lm}Q_{lm})/V_n$, where P_{lm} and Q_{lm} are the branch power flows and V_n is the nominal voltage of the distribution system. By aggregating the voltage drops along the unique path from the slack bus to node m (due to the radial structure), the total voltage deviation is expressed as:

$$\Delta V_m \approx \frac{1}{V_n} \sum_{l \neq m} \left(R_{lm} P_l^{inj} + X_{lm} Q_l^{inj} \right), \qquad (12)$$

s.t.

$$P_l^{inj} = P_l^{up} + \sum P_l^{PV} + \sum P_l^{WT} + \sum P_l^{dESS} - \sum P_l^{cESS} - P_l^{LD}, \forall l, \tag{13}$$

where R_{mj} and X_{mj} denote the mutual resistance and reactance (common path impedance) between nodes m and j, and P_j^{inj}, Q_j^{inj} represent the nodal power injections. Here, R_{mj} and X_{mj} capture the cumulative impedance along the shared path between nodes, which reflects the network topology in radial systems. P_m^{inj} is the inject active power at node m, Y_{mm} and Y_{lm} is the self-admittance and the mutual admittance between node l and m. P_m^{up}, P_m^{PV}, and P_m^{WT} is the active power from upper grid, PVs, and WTs. P_m^{LD} is the load. P_m^{dESS} and P_m^{cESS} is the discharging and charging active power of the ESSs which are connected to node m.

Although Q_j^{inj} physically contributes to the deviation, it is often assumed $\Delta Q \approx 0$ in active power impact studies, which simplifies the analysis by focusing on the dominant effect of active power on voltage variation. Since $R_{mj} \geqslant 0$ in radial topologies, the partial derivative $\partial V_m / \partial P_j^{inj} \approx R_{mj}/V_n$ is non-negative, proving that concurrent and co-directional injections from multiple nodes lead to a cumulative summation of voltage increments, thereby maximizing the total voltage deviation at the observation node.

The effectiveness of the full-agent coordination in MADSA lies in overcoming the action cancellation effect common in decentralized multi-agent systems. By aligning the perturbation vectors $\delta_{i,t}$ to target a specific physical boundary (e.g., a_i^{cmax} or $-a_i^{dmax}$), the attack transforms individual agent sensitivities into a collective systemic surge. This synchronization ensures that the cumulative voltage shift $\Delta V_m \approx \frac{1}{V_n} \sum_{l \neq m} (R_{lm} P_l^{inj} + X_{lm} Q_l^{inj})$ is maximized by ensuring all P_l^{inj} shifts are co-directional, effectively exploiting the radial topology of the ADN.

From Eqs. (12) and (13), we observe that the charging/discharging active power of the ESSs influences the voltage at the connected node m. Therefore, if the ESSs charge or discharge simultaneously, the voltage at node m will be maximally affected, assuming other factors, such as WTs, PVs, load, and active power from the upper power grid, remain constant.

Thus, aiming at the MADRL, the proposed MADSA considers the directed attack on all agents as follows:

MADSA$^-$ for full-agent attack. The attacker attempts to violate the lower bound voltage constraint. Namely, in the MADSA, for each agent $i \in \mathcal{A}_t^{att}$, it can be regarded as shifting the action as much as possible towards a_i^{cmax}.Then the function is set as $\mathcal{L}_{i,t}^{att} = \min||a'_{i,t} - a_i^{cmax}||$.

MADSA$^+$ for full-agent attack. Similarly, the attacker attempts to violate the upper bound voltage constraint. For each agent $i \in \mathcal{A}_t^{att}, \mathcal{L}_{i,t}^{att} = \min||a'_{i,t} - (-a_i^{dmax})||$.

Three different attack modes for single-agent attack and full-agent attack are discussed above. Next, MADSA solves the attack samples $\boldsymbol{o}'_{i,t}$ corresponding to the target attacked actions $\boldsymbol{a}'_{i,t}$ by minimizing the loss function $\mathcal{L}_{i,t}$, that is, the injected input of agent i.

Specifically, MADSA obtains attack samples based on an attack sample set $\mathcal{O}_i = \left\{ \delta_{i,t,j} |_{j=1}^{|\mathcal{O}_i|} \right\}$ of agent i, where each sample is computed through iteration. Donating the attack sample of agent i as $\boldsymbol{o}'_{i,t} = \boldsymbol{o}_{i,t} + \boldsymbol{\delta}_{i,t}$, then $\mathcal{L}_{i,t}^{att}$ is differentiable at $\boldsymbol{\delta}_{i,t}$. Taking MADSA^s as an example, the gradient of the differential of $\mathcal{L}_{i,t}$ is

$$\nabla_{\delta_{i,t}} \mathcal{L}_{i,t}^{att} = \nabla_{\delta_{i,t}} ||a'_{i,t} - a_{i,t}||. \tag{14}$$

Before solving each attack sample, randomly generate an initial perturbation $\delta_{i,t}^0$, and then iterate by Equation (15)-(17):

$$\boldsymbol{\delta}_{i,t}^{k+1} = \boldsymbol{\delta}_{i,t}^k + \alpha_{\mathcal{L}} \nabla_{\boldsymbol{\delta}_{i,t}} \mathcal{L}_{i,t}^{att} + \alpha_d \frac{1}{|\mathcal{O}_i|} \sum_{c=1}^{|\mathcal{O}_i|} (\boldsymbol{\delta}_{i,t,c} - \boldsymbol{\delta}_{i,t}^{k+1}), \tag{15}$$

where $\delta_{i,t}^k$ is the perturbation at k-th iteration, $\alpha_{\mathcal{L}} \nabla_{\delta_{i,t}} \mathcal{L}_{i,t}^{att}$ and $\alpha_d \frac{1}{|\mathcal{O}_i|} \sum_{c=1}^{|\mathcal{O}_i|} (\boldsymbol{\delta}_{i,t,c} - \boldsymbol{\delta}_{i,t}^{k+1})$ are terms about the loss function $\mathcal{L}_{i,t}^{att}$ and the distribution of the solution. $\alpha_{\mathcal{L}}$ and α_d is the step size of iteration. During each K-step iterative process of attack samples, the distribution term encourages the currently generated adversarial sample to deviate from those already included in the candidate set $\mathcal{O}_i$, thereby promoting diversity among the generated samples and leading to a more well-distributed candidate set. This design is intended to prevent repeated getting stuck in a local optimum when searching for adversarial samples.

Considering that $\mathcal{O}_i = \emptyset$ before solving the first sample, we specially stipulate that the term of the distribution is the $\mathbf{0}$ at this time. After each iteration, $\delta_{i,t}^k$ is projected to the allowed range:

$$\boldsymbol{\delta}_{i,t}^{k+1} = \mathrm{Proj}\left(\boldsymbol{\delta}_{i,t}^{k+1}, \epsilon\right). \tag{16}$$

Assuming the iteration converges at step K, the resulting perturbation $\boldsymbol{\delta}_{i,t}^K$ is added to the candidate set $\mathcal{O}_i$. This process repeats until the set reaches a predefined size $|\mathcal{O}_i| = Z$. Once the candidate set is fully constructed, the final attack sample $\delta_{i,t}$ is selected via the following sampling mechanism:

$$\delta_{i,t} \sim \mathcal{P} = \mathrm{softmax}\left(\mathcal{L}_{i,t}^{att} |_{a'_{i,t} = \mu_{\pi_i}(\boldsymbol{o}_{i,t} + \boldsymbol{\delta}_{i,t,c})}, \forall\, \boldsymbol{\delta}_{i,t,c} \in \mathcal{O}_i \right), \tag{17}$$

where $\mathcal{P}$ denotes the probability distribution derived by applying the softmax function to the loss values of all candidates in $\mathcal{O}_i$.

Algorithm 1 MADSA Attack Procedure

Input: Joint policy $\Pi = \pi_1 \times \cdots \times \pi_n$, observation $\boldsymbol{o}_{i,t}$
Output: Adversarial sample $\boldsymbol{o}'_{i,t}$
for $t = 1$ **to** T **do**
  for $i \in \mathcal{A}$ **do**
    if agent i is attacked **then**
      Initialize candidate set $\mathcal{O}_i \leftarrow \emptyset$;
      for $z = 1$ **to** Z **do**
        Randomly initialize $\boldsymbol{\delta}^0_{i,t}$;
        for $k = 0$ **to** $K-1$ **do**
          $\boldsymbol{\delta}^{k+1}_{i,t} = \boldsymbol{\delta}^k_{i,t} + \alpha_{\mathcal{L}} \nabla_{\delta_{i,t}} \mathcal{L}^{att}_{i,t} +$
          $\alpha_d \frac{1}{|\mathcal{O}_i|} \sum_{\delta_{i,t,c} \in \mathcal{O}_i} \left(\delta_{i,t,c} - \boldsymbol{\delta}^k_{i,t}\right)$
          $\boldsymbol{\delta}^{k+1}_{i,t} \leftarrow \mathrm{Proj}(\boldsymbol{\delta}^{k+1}_{i,t}, \epsilon)$
        end for
        $\mathcal{O}_i \leftarrow \mathcal{O}_i \cup \{\boldsymbol{\delta}^K_{i,t}\}$;
      end for
      Compute selection probability: $\delta_{i,t} \sim \mathcal{P} =$
      $\mathrm{softmax}\left(\mathcal{L}^{att}_{i,t}\big|_{a'_{i,t}=\mu_{\pi_i}(\boldsymbol{o}_{i,t}+\boldsymbol{\delta}_{i,t,c})}, \forall\, \boldsymbol{\delta}_{i,t,c} \in \mathcal{O}_i\right)$,
      Sample $\delta_{i,t} \sim \mathcal{P}$;
      $\boldsymbol{o}'_{i,t} \leftarrow \boldsymbol{o}_{i,t} + \boldsymbol{\delta}_{i,t}$;
    end if
  end for
end for

By iteratively executing Equations (14)–(17), the objective is to minimize the attack loss $\mathcal{L}^{att}_{i,t}$. This ensures that the adversarial observation drives the agent's actual output $a'_{i,t}$ as close as possible to the attacker's target action. Consequently, the optimal attack sample for agent i is determined as $\boldsymbol{o}'_{i,t} = \boldsymbol{o}_{i,t} + \boldsymbol{\delta}_{i,t}$. Depending on the attack mode (single-agent or full-agent), the joint attack sample for the entire MADRL system is then formed by combining the samples from all targeted agents. The detailed execution of MADSA is summarized in Algorithm 1.

3 Defense Mechanisms of MADRL Based ADNs

In response to the proposed multi-agent adversarial attacks against ADNs, this section introduces a defense method named MAGLD. As shown in Figure 3, this method performs gradient leveling of the loss function near the interaction sequence to form local saddle points of the gradient, thereby reducing the models' sensitivity to input perturbations and enhancing their adversarial robustness.

3.1 Gradient Leveling Based Adversarial Defense Model

Assuming a multi-agent system with a loss function $\mathcal{L}(\theta)$, where θ represents model parameters. In training of RL, denoting θ_i as the model parameters of agent i, the policy

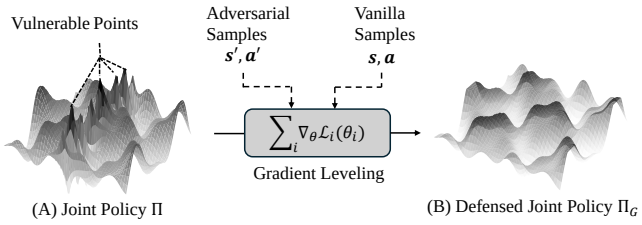


Fig. 3 A conceptual diagram of the attack-defense mechanism. (A) The attack strategy is to find the vulnerable points in the joint policy Π where the joint gradient is steep, so that a slight change in the observation causes a significant deviation in the joint decision. (B) The defense strategy is to level the gradient at these steep points as much as possible using gradient leveling regularization.

gradient for each agent i can be defined as

$$\nabla_{\theta_i} \mathcal{L}_i(\theta_i) = \mathbb{E}_{\boldsymbol{o}_i \sim d_{\pi_i}} \left[\nabla_\theta \log_{\pi_i}(a_i \mid \boldsymbol{o}_i; \theta_i)\, r_i(\boldsymbol{o}_i, a_i)\right], \tag{18}$$

where d_{π_i} is the state distribution under policy π_i. During training, the model may exhibit a linear response to certain inputs, causing significant changes in actions due to state changes. At this time, adversarial samples can cause significant output changes through small perturbations.

Definition 3.1: Gradient leveling regularization. Gradient leveling regularization is defined as the squared Euclidean norm of the summed gradients of each agent's loss function $\mathcal{L}_i(\theta_i)$ with respecting to the model parameters θ_i:

$$\sum_i ||\nabla_\theta \mathcal{L}_i(\theta_i)||_2^2, \forall\, i \in \mathcal{A}. \tag{19}$$

Equation (19) is used to smooth the gradients of all agents, reducing the model's sensitivity to input perturbations, thereby improving its adversarial robustness and stability. By introducing this regularization term into the loss function, the parameter updates during optimization will be smoother, reducing the risk of overfitting and enhancing the generalization ability of the multi-agent system in complex and adversarial environments.

The meaning of using the regularization term $\sum_i \nabla_{\theta_i} \mathcal{L}_i(\theta_i)$ is that by flattening the gradients of all agents, the gradient magnitude of each agent's loss function is reduced. This helps make parameter updates smoother during optimization, avoiding instability caused by large gradients. Physically, this means that the model is no longer sensitive to small perturbations in the input data, thereby improving the system's adversarial robustness and stability. Especially in the face of adversarial attacks, this regularization term makes small input perturbations less likely to cause significant output changes, enhancing the reliability and anti-interference capability of the multi-agent system in complex and adversarial

environments.

Lemma 3.1. In a multi agent system, introducing the gradient leveling regularization term $\lambda \sum_i ||\nabla_{\theta_i} \mathcal{L}_i(\theta_i)||_2^2$, the loss function of the multi agent system is defined as

$$\mathcal{L}_{smooth}(\theta) = \mathcal{L}(\theta) + \lambda \sum_i ||\nabla_\theta \mathcal{L}_i(\theta_i)||_2^2. \quad (20)$$

By optimizing $\mathcal{L}_{smooth}(\theta)$, that is, reducing the squared Euclidean norm of distribution gradient $\nabla_{\theta_i} L_i(\theta_i)$ under $\boldsymbol{s} \sim d_\pi$, the adversarial robustness of agent model is improved, where λ is the regularization coefficient.

However, $\nabla_{\theta_i} \mathcal{L}_i(\theta_i)$ is difficult to calculate directly. Therefore, this paper proposes Lemma 3.2 to solve this problem.

Lemma 3.2. It can approximate the gradient leveling term by introducing the adversarial sample loss $\mathcal{L}_i(\theta_i, \boldsymbol{o}_i') - \mathcal{L}_i(\theta_i, \boldsymbol{o}_i)$, thereby reducing the model's sensitivity to input perturbations.

Lemmas 3.1 and 3.2 together demonstrate that introducing adversarial samples into the training process can enhance the adversarial robustness of reinforcement learning agents. Lemma 3.1 formalizes the idea through a gradient-leveling regularization term, while Lemma 3.2 provides a practical way to approximate this term using adversarial sample losses. Thus, the proposed approach not only improves robustness for single-agent RL but can also be generalized to multi-agent systems.

By optimizing $\lambda \mathcal{L}_i(\theta_i, \boldsymbol{o}_i') + (1-\lambda)\mathcal{L}_i(\theta_i, \boldsymbol{o}_i)$, the sensitivity of the model to input perturbations can be reduced.

The mitigation capability of MAGLD is rooted in its ability to flatten the policy response to directed shifts. Unlike detection-based defenses that may suffer from false positives or latency, gradient leveling provides a proactive mitigation by ensuring that the mapping from observations to actions is locally stable. This architectural robustness ensures that even if a coordinated attack succeeds in shifting the input data toward a malicious direction, the resulting action deviation is insufficient to cause the system to exit its safe operational envelope.

3.2 Training and Application of Defense

Ulteriorly, this section provides a detailed implementation of training and application of defense.

Taking MADDPG as an example, it consists of two network sets: the main and target networks, with each set containing both an actor network and a critic network. Let θ_i^μ and $\theta_i^{\mu'}$ be the parameters of main critic network and target critic network, the main critic network is updated by $\mathcal{L}_{smooth} \approx \lambda \mathcal{L}_i(\theta_i, \boldsymbol{o}_i') + (1-\lambda)\mathcal{L}_i(\theta_i, \boldsymbol{o}_i)$. Here $L(\theta_i, \boldsymbol{o}_i)$ is temporal difference (TD) error, which is formulated as:

$$L(\theta_i, \boldsymbol{o}_i) = \mathbb{E}[r_i + \gamma Q_{\theta_i^{\mu'}}(\boldsymbol{o}_{i,t+1}, \mu_\Pi(\boldsymbol{s})) - Q_{\theta_i^\mu}(\boldsymbol{o}_{i,t}, a_{i,t})], \quad (21)$$

where $Q_{\theta_i^\mu}(\boldsymbol{o}_i, a_{1:N})$ and $Q_{\theta_i^{\mu'}}(\boldsymbol{o}_{i,t+1}, \mu_\Pi(\boldsymbol{s}))$ is the approximation of value function output by θ_i^μ and $\theta_i^{\mu'}$. This loss function reflects the gap between the current value estimate and the target value. Minimizing it updates the value function to better satisfy the Bellman equation, thereby improving the policy. $\mathcal{L}_i(\theta_i, \boldsymbol{o}_i')$ is calculated by similar way in the distribution of adversarial samples, that is, the attacked sample $(\boldsymbol{o}_i', a', r', \boldsymbol{o}_{i,t+1}')$ is used to replace the original sample.

Then, the main actor network θ_i^Q is used to map the state to a deterministic action and is updated by the sampled policy gradient, which is formulated as

$$\nabla_{\theta_i^\mu} J^\mu(\mu_{\theta_i^\mu}) \approx \lambda \mathbb{E}\left[\nabla_{\theta_i^\mu} \mu_i(\boldsymbol{o}_i) \nabla_{a_i} Q_i^\mu(\boldsymbol{o}_i, a_i)\right] + (1-\lambda)\mathbb{E}\left[\nabla_{\theta_i^\mu} \mu_i(\boldsymbol{o}_i') \nabla_{a_i'} Q_i^\mu(\boldsymbol{o}_i', a_i')\right]. \quad (22)$$

Finally, the target networks are soft updated, which refers to gradually updating the parameters of target networks by blending them with the parameters of main networks using a small update rate.

It is worth mentioning that although adversarial gradient leveling regularization improves adversarial robustness, adversarial training still slightly reduces scheduling accuracy. In implementation, considering the low frequency of FDI attacks, the original model strategy and the gradient leveling regularization model are integrated for decision making, which can be expressed as:

$$\boldsymbol{a}_t = \begin{cases} \mu_\Pi(\boldsymbol{s}_t), & \mathcal{R}(\boldsymbol{s}_t, \mu_\Pi(\boldsymbol{s}_t)) = 1, \\ \mu_{\Pi^G}(\boldsymbol{s}_t), & \mathcal{R}(\boldsymbol{s}_t, \mu_\Pi(\boldsymbol{s}_t)) = 0, \end{cases} \quad (23)$$

where Π^G is joint policy of agents trained by the MAGLD.

Although introducing gradient leveling regularization improves adversarial robustness, it negatively impacts reward stability, decreasing the convergence ability of MADRL algorithms. In practical applications, a balance between adversarial robustness and convergence performance must be found. Therefore, the value of sample concentration λ is discussed in Section. IV.

In practical applications, MAGLD should follow the "Sim-to-Real" paradigm of DRL [30]. Specifically, MAGLD's adversarial training is also a DRL training process, so in the long-term operation after deployment, it can be considered to ensure the stability of the strategy while adapting to changes in the environment through fine-tuning.

4 Case Study

To verify the performance of the proposed attack method MADSA and defense method MAGLD[①], the case studies are displayed by the following parts:

A. Continuous attacks against full-agent in MADRL;

B. Single-step attack against full-agent and single-agent in MADRL;

C. Defense simulations;

D. Generalization experiments;

E. Simulation and Analysis based on a Real World ADN.

Unless otherwise specified (e.g., to demonstrate the applicability of the proposed attack to different DRL training algorithms), this paper uses the model trained with MADDPG and $\lambda = 0.2$ for simulation (The relative best setting verified in Section IV.D). The multi-agent training is well-established based on a widely used neural network structure–Long-Short-Term Memory (LSTM). The case study utilizes the IEEE 33-node ADN, which includes WTs and PVs. The rated power of all WTs and PVs is set to 400 kW. The ADN contains four agents representing four ESSs, which are randomly connected to nodes 7, 15, 18, and 29. The ESSs have a maximum charging and discharging active power of 2 MW and an energy capacity of 7.7 MWh. At the beginning of each day, the initial state of charge of the ESSs is set to 50%. It is assumed that the PVs, WTs, and load fluctuate proportionally, with the fluctuation curve shown in reference [33]. About the Dec-PMODP, the voltage deviation threshold β is 7% according to the standard of State Grid Corporation of China and the penalty coefficient ρ is 100.

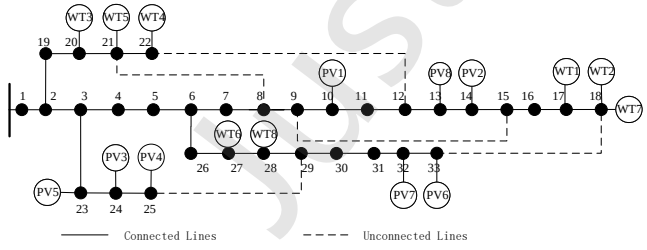


Fig. 4 IEEE 33-node ADN with WTs and PVs.

In the vanilla MADRL training and adversarial training, the Critic networks adopt a network architecture consisting of Dense(64), LSTM(128), LSTM(128), and Dense(64), while the Actor networks are composed of three fully connected layers with 64 neurons each. The learning rates of both the Actor and the Critic are set to 10^{-3}. The batch size is 64, the replay buffer size is 10^6, and the soft update rate is 10^{-3}. The exploration noise follows an adaptive Gaussian distribution, which is uniformly decayed from $\mathcal{N}(0, 10)$ to $\mathcal{N}(0, 10^{-2})$.

In the adversarial training process of MAGLD, the adversarial samples are composed of three types, namely MADSA^s, MADSA$^+$, and MADSA$^-$. In our implementation, these three types of adversarial samples are equally sampled, each accounting for one-third of the total adversarial samples, which provides a simple and practical setting to ensure diversity while maintaining reproducibility.

4.1 Continuous Attacks Against Full-Agent in MADRL

We explored the effect of continuous MADSA attacks on all-agent DRL with varying attack amplitudes (0.05, 0.1, 0.2 respectively) here. Table 2 presents the objective and corresponding ADNs losses under these attack amplitudes, along with existing attack methods:

- STD: Vanilla MADRL model without attack.
- Noise: Noise attack [21].
- FGSM: FGSM attack [19].
- PGD: Projected Gradient Descent attack [31].
- CW: Carlini Wagner attack [32].
- Universal: Universal attack [33].
- MADSA: Proposed attacks methods.

Figure 5 further shows the voltage distribution statistics of each node when MADSA selects different attack directions with 0.05 attack amplitude, displayed in the form of a box plot.

Table 2 Results Under Different Continuous Attacks Against Full-agent

	F^{obj}			Power loss (MWh)		
$\epsilon =$	0.05	0.1	0.2	0.05	0.1	0.2
STD		**9.375**			**1.372**	
FGSM [21]	14.014	19.697	31.132	1.829	2.291	3.416
Noise [19]	10.492	10.928	13.583	1.395	1.472	1.838
CW [32]	16.939	25.721	39.790	1.859	3.001	5.671
Universal [33]	16.629	26.241	39.785	1.948	3.229	5.277
PGD [31]	17.311	26.220	39.156	2.043	3.490	5.965
MADSA^s	17.556	26.295	39.748	1.898	3.005	5.263
MADSA$^+$	23.447	40.814	44.308	2.387	7.119	8.243
MADSA$^-$	22.959	31.007	54.808	2.090	3.663	8.638

As shown in Table 2, the degradation of ADNs increases with higher attack amplitudes, reflected in the rising of the F^{obj} value and power loss. Even a MADSA continuous attack with an amplitude of 0.05 leads to significant losses of 38.34%, 73.98%, and 52.33% across different attack directions compared to the Std model. Furthermore, MADSA

① The source code, trained models and more information are available at https://github.com/zhuocend/MADSA.

induces greater power losses and the F^{obj} than FGSM and noise attacks of the same amplitude, demonstrating a more effective attack.

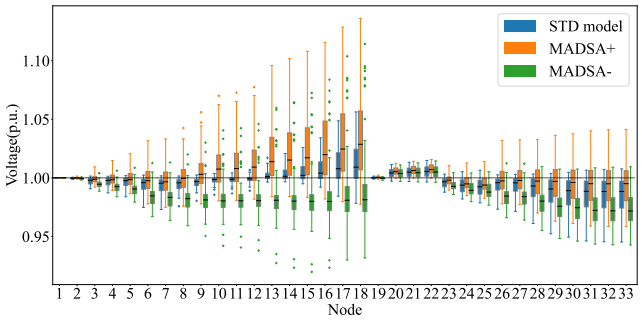


Fig. 5 Voltage distribution statistics of each node when MADSA selects different attack directions with 0.05 attack amplitude.

Further analysis and comparison of the impact of the three basic MADSA models on the attack shows that the MADSA^s model considers the maximum deviation of the single-agent action. However, for the full-agent attack, it cannot ensure the consistency of each agent's action direction, leading to potential internal cancellation, resulting in lower reward values and losses compared to MADSA$^-$ and MADSA$^+$. This lack of coordination explains why traditional adversarial methods like FGSM often yield sub-optimal attack results when applied to the complex, interactive environment of MADRL. From the Figure 5, MADSA$^+$ and MADSA$^-$ cause significant upward and downward overall deviations in the ADNs voltage, demonstrating the effectiveness direction choosing of MADSA against the full-agent DRL.

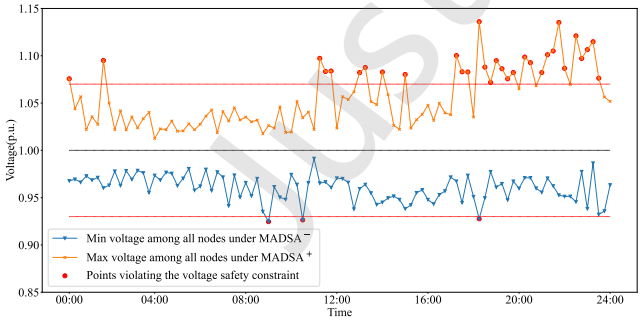


Fig. 6 Voltage extrema of all nodes over time under proposed attack method.

Figure 6 depicts the voltage extrema of all nodes in the ADN over time under MADSA$^-$ and MADSA$^+$ attacks. It can be clearly seen that, although the attacks are not always successful at every moment—i.e., voltage limit violations do not occur continuously—there are still numerous instances of voltage violations, which fully demonstrates the effectiveness of the proposed attack method. At the same time, an interesting phenomenon can be observed: in the latter part of the time series, voltage violations occur more frequently. This is because the deviations caused by the attacks accumulate and propagate over time, making them more likely to manifest in the ADN system. Furthermore, by treating PGD as an iterative adversarial attack baseline, it can be observed that $MADSA^S$ holds only a slight advantage over PGD. In contrast, both $MADSA^-$ and $MADSA^+$ are capable of increasing voltage fluctuations to a much greater extent. This demonstrates that the directed shift within MADSA is the main reason for the enhanced effectiveness of the full-agent attack.

Considering the black-box setting as being more representative of practical engineering scenarios, we conduct black-box continuous attack experiments in which the attacker has no access to any internal system information and can only observe the model inputs and outputs. In these black-box experiments, we train the same number of transfer models via behavior cloning based on the inputs and outputs of the MADRL agents, and then generate adversarial samples using the behavior-cloned models.

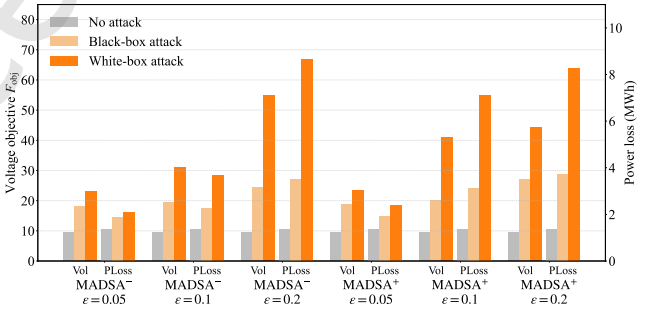


Fig. 7 Comparison of White-box and Black-box Attacks using MADSA.

Figure 7 illustrates the performance comparison between black-box and white-box attacks under the two attack modes, $MADSA^-$ and $MADSA^+$. It can be observed that when $\epsilon = 0.05$, the black-box attack achieves approximately 60%–70% of the voltage objective increase and power loss increase induced by the corresponding white-box attack. This indicates that although the attack effectiveness is reduced under the black-box setting compared with the white-box case, the attack still poses a non-negligible threat, thereby revealing the vulnerability of MADRL-based ADN dispatching.

4.2 Single-step Attack Against Full-Agent and Single-Agent in MADRL

Given the challenge of continuous attacks for 24 hours without detection, attackers often opt for single-step attacks at crucial moments in ADNs. This approach disrupts the overall operation of the ADNs. Consequently, Case 2 focuses on

single-step attack and evaluates their impact on both full-agent and single-agent MADRL systems.

Figure 8 shows the voltages of ADN at 18:15, before and after single-step MADSA, noise, and FGSM attacks. At this time, the conditions are 75.62% WTs, 0% PVs, and 100% load.

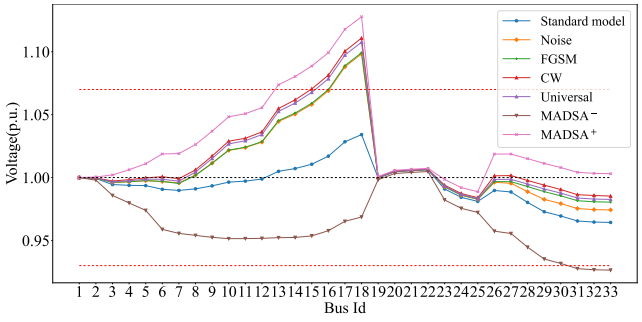


Fig. 8 Voltage distribution after the different single-time attacks at t=18:15.

As shown in Figure 8, before the attack, the voltage of each node was relatively stable, fluctuating between 0.97 and 1.03, with no node voltage exceeding the limit. After being subjected to a Single-step attack, whether it is MADSA$^+$, FGSM, or Noise attack, some nodes exceeded the upper limit (1.07 p.u.). However, the proposed MADSA$^+$ has a wider impact range (i.e., the number of nodes exceeding overvoltage are the largest) and a higher degree of deterioration (i.e., the amplitude of exceeding the limit is the largest). This indicates that the proposed attack algorithm is the most powerful in Single-step attacks. Furthermore, only the proposed MADSA$^-$ can cause the voltage of the ADN to drop, with nodes 31, 32, and 33 exceeding the lower limit (0.93 p.u.). This further demonstrates that in the Full-agent DRL, the proposed MADSA can autonomously choose to attack the voltage upward or downward, making the attack strategy more complex and effective. It should be noted that the timing of the Single-step attack should be combined with the ADN state, i.e., choosing the critical moment when the system is most vulnerable will lead to a more obvious impact of the attack.

Then we test single-step attacks against different agents in MADRL at t=18:15 with 0.05 amplitude. The results are shown in Table 3.

From Table 3 we can see that after different agents are attacked, the impact on ADNs varies, reflected in the maximum and minimum voltage changes. This is understandable because the FDIs attack modifies the observed quantity injected into the agent, indirectly affecting the decision of the agent, i.e., the charging and discharging behavior of ESSs. The impact of this behavior on ADNs is naturally constrained by the actual physical power flow. A specific analysis shows that only attacking ESS 1 at node 18 will cause the overvoltage, while attacking other agents cannot result in overvoltage. This is due to the location of the ESS. For the ESS 1, where is at node 18, its position at the endpoint easily causes the voltage to rise when discharge. Additionally, at t=18:15, in the single-step attack on ESS 1, FGSM and Noise cannot cause the limit to exceed, which verifies the strength of the MADSA^s attack. Compared to the relatively advanced CW and Universal attacks, MADSA achieves similar performance. Finally, it should be noted that unlike continuous attacks, using MADSA$^-$ to implement single-step attacks at a specific time makes it difficult to cause the voltage to fall below the lower limit.

Table 3 Results of attacking different single-agent MADRL under 0.05 amplitude attack at time 18:15.

	ESS 1 (18)		ESS 2 (7)		ESS 3 (29)		ESS 4 (15)	
	Min Vol	Max Vol	Min Vol	Max Vol	Min Vol	Max Vol	Min Vol	Max Vol
FGSM [21]	0.974	1.069	0.944	1.017	0.982	1.043	0.968	1.047
Noise [19]	0.976	1.056	0.958	1.030	0.968	1.037	0.972	1.033
CW [32]	0.977	**1.099**	0.948	1.014	0.985	1.046	0.971	1.055
Universal [33]	0.978	**1.097**	0.946	1.015	0.982	1.042	0.967	1.058
PGD [31]	0.976	**1.098**	0.948	1.014	0.982	1.046	0.966	1.052
MADSA^s	0.978	**1.098**	0.943	1.014	0.981	1.046	0.966	1.053

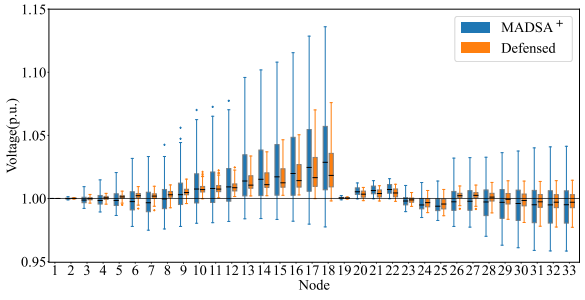
4.3 Defense Experiments

To show the defense effect, Figure 9 presents the defense results (box plot of voltage) against continuous attacks by MADSA$^+$ and MADSA$^-$ attack.

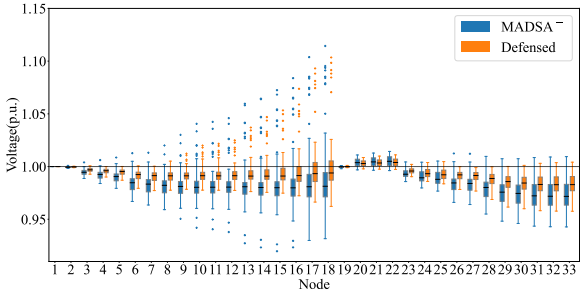
As shown in Figure 9(a), continuous attacks by MADSA$^+$ cause abnormal voltage values (deviation from the median of the box plot), representing the overvoltage of nodes in ADNs. After implementing the defense strategy, the voltage becomes more stable, shifting downward overall and eliminating the overvoltage issue. Similarly, Figure 9(b) illustrates that after continuous attacks by MADSA$^-$, some nodes (e.g., nodes 13-16) fall below the lower limit. Post-defense, the overlimit issue is resolved, and the node voltages shift upward overall, mitigating the impact of MADSA$^-$. In summary, the defense strategy effectively counters continuous attacks by MADSA.

Finally, we train different defense models by the following methods:

- NoiseT: Trained with Noise examples [21].
- FGSMT: Trained with FGSM examples [34].
- FAST: An improve defense method of FGSMT [35].



(a) Defense result against continuous attacks by MADSA$^+$.



(b) Defense result against continuous attacks by MADSA$^-$.

Fig. 9 Voltage box plot of the defense results.

- TRADES: TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization [36].
- SA-PPO: State-Adversarial Proximal Policy Optimization [37].
- ATLA-PPO: Alternating Training of Learned Adversaries Proximal Policy Optimization [38].
- Ours: Trained with the proposed attack method MADSA and defense method MAGLD.

We then test these models against different single-step attacks, and the results are shown in Table 4. It should be noted that to verify the effectiveness of the defense, we perform single-step attacks 96 times, and the maximum/minimum voltage under attacking.

As shown in Table 4, the proposed defense (Ours) described in the paper effectively defends against single-step attacks, including MADSA, FGSM, and Noise, preventing voltage from exceeding limits. However, the adversarial sample (NoiseT) using FGSM struggles to defend against the MADSA attack described in the article. MADSA$^+$ causes the voltage to exceed the upper limit, MADSA$^-$ causes it to drop below the lower limit, and MADSA^s attacks on a single-agent lead to overvoltage. Similarly, the adversarial sample (FGSMT and FAST) using Noise is ineffective against MADSA and FGSM attacks. This is understandable: as the attack strength of Noise, FGSM, and MADSA increases, defense difficulty increases accordingly. Training with higher-strength adversarial samples can improve defense effectiveness, but it should be noted

Table 4 Defense results (against single-step attack) of different attacking examples.

Attack method and mode		MADSA −	MADSA +	MADSA s	PGD	FGSM	Noise
NoiseT [21]	Min vol	**0.920**	0.971	0.952	0.950	0.951	0.957
	Max vol	1.015	**1.148**	**1.115**	**1.112**	**1.112**	1.063
	Vio Cnt*	4	21	8	7	5	0
FGSMT [34]	Min vol	**0.927**	0.971	0.946	0.950	0.944	0.957
	Max vol	1.017	**1.137**	**1.120**	**1.117**	1.057	1.059
	Vio Cnt	2	25	10	10	0	0
FAST [35]	Min vol	**0.927**	0.970	0.948	0.949	0.946	0.954
	Max vol	1.016	**1.141**	**1.119**	**1.121**	1.059	1.059
	Vio Cnt	2	16	6	10	0	0
TRADES [36]	Min vol	0.938	0.982	0.952	0.951	0.953	0.959
	Max vol	1.021	1.070	1.063	1.065	1.063	1.057
	Vio Cnt	0	0	0	0	0	0
SA-PPO [37]	Min vol	0.932	0.975	0.952	0.948	0.951	0.958
	Max vol	1.015	**1.092**	1.068	1.068	1.060	1.057
	Vio Cnt	0	3	0	0	0	0
ATLA-PPO [38]	Min vol	0.932	0.980	0.944	0.950	0.952	0.955
	Max vol	1.024	**1.077**	1.066	1.064	1.054	1.061
	Vio Cnt	0	1	0	0	0	0
Ours	Min vol	0.939	0.973	0.949	0.953	0.958	0.948
	Max vol	1.026	1.064	1.063	1.062	1.056	1.060
	Vio Cnt	0	0	0	0	0	0

*The rows with head "Vio cnt" represent the number of violations in a daily dispatching under corresponding attack and defense method.

that the convergence of the defense model may be affected after integrating a strong adversarial sample generation strategy. As adversarial PPO variants, SA-PPO and ALTA-PPO exhibit significantly better defense performance than adversarial training based solely on FGSM, yet they still experience voltage upper-limit violations under the MADSA$^+$ attack. As an advanced defense method, TRADES also ensures that the voltage does not exceed the limit, but it can be observed that the proposed defense has a slight advantage over TRADES in executing multi-agent attacks. Specifically, MADSA$^-$ is 0.938 vs 0.939 and MADSA$^+$ is 1.064 vs 1.070.

For details, see the comparison described in Figure 9.

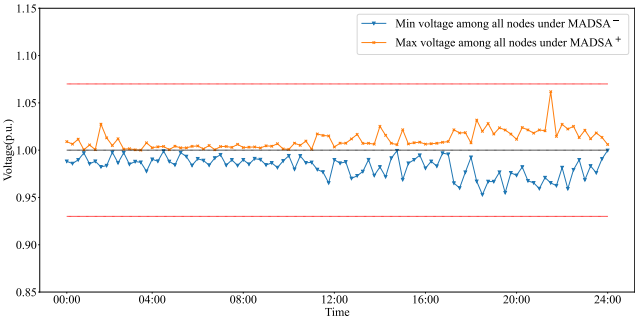


Fig. 10 Voltage extrema of all nodes over time of proposed defense method.

Figure 10 illustrates the voltage extrema of all nodes in the ADN under $MADSA^-$ and $MADSA^+$ attacks for the

defense model. It can be observed that, compared to the case without defense, voltage limit violations do not occur at any time, although the voltage amplitude slightly increases at certain moments.

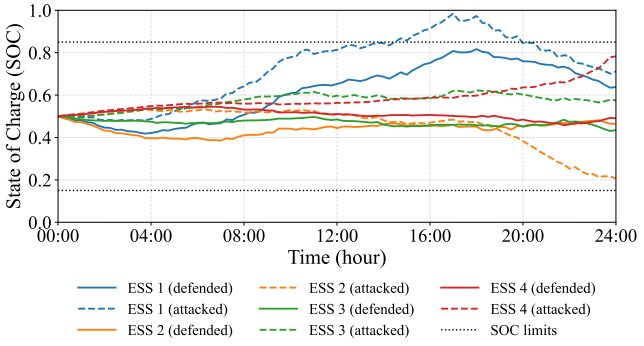


Fig. 11 Comparison of SOC trajectories under $MADSA^{-}$ attacks with and without defense.

Taking $MADSA^{-}$ as an example, Figure 11 illustrates the SOC trajectories of ESSs 1–4. It can be observed that, under attack, some ESSs may violate the prescribed SOC limits, although this behavior does not occur for all units. In contrast, after applying the proposed defense method, the SOC constraints are strictly satisfied, and no associated penalty is incurred.

4.4 Generalization Experiments

The generalization of proposed attack and defense methods are verified from the DRL algorithm and the adversarial samples concentration parameter λ. For the MADRL algorithm, we test three representative MADRL algorithms: MADDPG [39], MAPPO [40], and QMIX [41], with agent counts ranging from 4 to 8. For the adversarial samples concentration parameter λ, we tried multiple parameter settings at equal intervals, that is $\lambda = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6$.

The results of generalization experiments shown in Table 5. From Table 5, we first compare the attack and defense effects with different numbers of agents under the same training algorithm. It is evident that as the number of ESSs increases, the severity of voltage limit violations (upper and lower) caused by the attack worsens. This is understandable because with more agents, there are more targets for the attack algorithm, resulting in more significant degradation of the ADNs due to their collaboration. However, after applying MAGLD, these attacks can be effectively countered, and the defense model remains capable of converging despite the increased number of agents. This demonstrates the generalization of the attack and defense strategy across different numbers of agents and topologies. Additionally, it is found that the three common

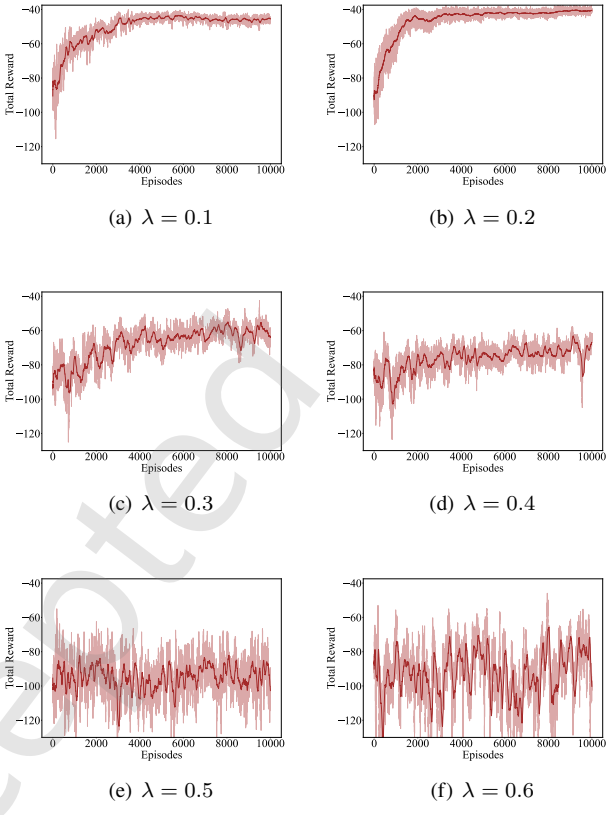


Fig. 12 Reward curves with different adversarial samples concentration parameter λ.

multi-agent training algorithms can be effectively attacked and defended, demonstrating the generalization of the strategy across different algorithms.

Finally, it can be clearly observed from Figure 12 that as λ increases, the convergence deteriorates. When λ does not exceed 0.2, adversarial training exhibits good convergence, while no convergence is observed within 10,000 episodes when λ exceeds 0.5. Therefore, the conclusion that can be drawn from Figure 12 and Table 5 is that a balance should be struck between incorporating more adversarial samples to enhance adversarial robustness and adding fewer samples to ensure convergence, with $\lambda = 0.2$ being the optimal choice.

4.5 Simulation and Analysis based on Real World Data

The previous experiments fully validated the effectiveness and generalization of the proposed attack method MADSA and defense method MAGLD. This section provides simulations conducted on real-world data.

Figure 13 shows the distribution network topology used in this section. It represents an ADN in a city in Guangdong, China, covering part of the city. The network consists of 3

Table 5 Attack and defense results of different number of agents, different MADRL training algorithm, different neural network, different λ.

			STD			Attacked			Defended		
			Objective	Min vol	Max vol	Objective	Min vol	Max vol	Objective	Min vol	Max vol
Generalization of DRL algorithms	4ESSs	MADDPG	9.375	0.955	1.054	23.444	0.906	1.149	11.132	0.939	1.064
		MAPPO	9.405	0.946	1.052	23.434	0.911	1.155	11.155	0.941	1.065
		QMIX	9.376	0.936	1.054	23.429	0.915	1.154	11.131	0.937	1.067
	6ESSs	MADDPG	9.050	0.955	1.057	25.467	0.906	1.149	11.016	0.938	1.068
		MAPPO	9.155	0.944	1.065	25.435	0.904	1.153	11.056	0.933	1.073
		QMIX	9.091	0.948	1.061	25.525	0.909	1.159	11.020	0.936	1.068
	8ESSs	MADDPG	8.803	0.940	1.053	27.377	0.899	1.168	10.923	0.942	1.063
		MAPPO	8.781	0.940	1.055	27.631	0.901	1.169	10.929	0.935	1.064
		QMIX	8.788	0.937	1.053	27.367	0.910	1.168	10.935	0.947	1.067
Generalization of λ in adversarial training	$\lambda = 0.1$		9.750	0.954	1.055	-	-	-	13.298	0.952	1.064
	$\lambda = 0.2$		9.946	0.952	1.062	-	-	-	11.132	0.949	1.064
	$\lambda = 0.3$		11.080	0.951	1.071	-	-	-	12.192	0.937	1.079
	$\lambda = 0.4$		19.345	0.933	1.118	-	-	-	25.880	0.910	1.162
	$\lambda = 0.5$		27.730	0.914	1.150	-	-	-	29.090	0.891	1.179
	$\lambda = 0.6$		28.422	0.915	1.163	-	-	-	29.175	0.895	1.203

*The objective, maximum voltage and minimum voltage in the table are recorded as the worst values under proposed attack strategies.

transformers and 60 nodes. The types of overhead conductors and cables used for the lines in the ADN are shown in Table 6. In this ADN, ESSs, PVs, WTs and Loads are connected to the following nodes:

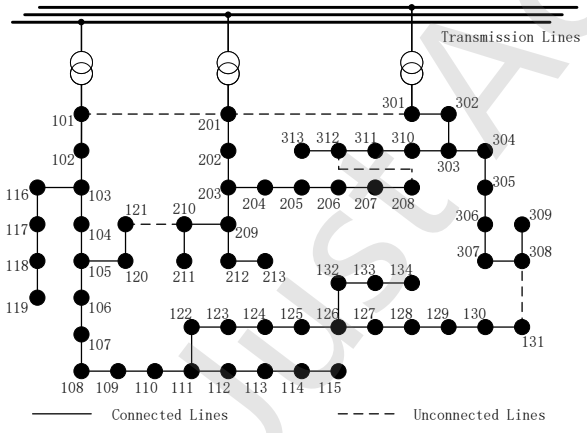


Fig. 13 Topology of the real world ADN.

- ESSs: 115, 118, 124, 133, 207, 209, 302, 306.
- PVs: 101, 103, 107, 115, 116, 117, 118, 120, 127, 131, 133, 134, 205, 206, 213, 302, 304.
- WTs: 102, 103, 106, 109, 115, 115, 115, 128, 130, 134, 302, 304, 312, 312, 313.
- Loads: 102 to 134, 202 to 213, 302 to 313.

Based on the above data, we provide cross-validation of all attack and defense methods on the real ADN topology, as shown in Table 7.

When no defense is applied, $MADSA^{+}$ can maximize the deterioration of the objective function by up to 182.52%. According to the definition of F^{obj}, the deterioration of the objective function implies that the average magnitude of voltage deviations also increases by the same factor.

Table 6 The types of overhead conductors and cables used in the lines of the real-world ADN.

Type	Lines
JL/G1A-240/30	116-117, 128-129, 129-130, 203-204
JL/LB1A-240/30	105-120, 117-118, 118-119, 120-121, 126-127, 310-311, 311-312
JL/LB20A-630/45	101-102, 102-103, 103-116, 111-122, 209-212, 212-213, 301-302, 302-303, 303-304, 303-310, 307-308, 308-309
YJLW03 1*630	103-104, 104-105, 105-106, 106-107, 107-108, 108-109, 109-110, 110-111, 111-112, 112-113, 113-114, 114-115, 122-123, 123-124, 124-125, 125-126, 126-132, 127-128, 130-131, 132-133, 133-134, 201-202, 202-203, 304-305, 305-306, 306-307
LGJ-300	203-209, 204-205, 205-206, 206-207, 207-208, 209-210, 210-211, 312-313

When defense measures are implemented, the proposed defense method MAGLD still achieves the best performance, particularly against the powerful multi-agent attack method MADSA. Regarding extreme voltage values, the proposed defense method MAGLD ensures that voltages at any time remain within constraints. Specifically, the minimum voltage of 0.936 is recorded at node 131, while the maximum voltage of 1.067 is observed at node 134. Among all defense algorithms,

Table 7 F^{obj} Under Continuous Attack and Defense in real ADN Topology.

Attack method	Defense method					
	NoDEF*	NoiseT [21]	FGSM [34]	T FAST [35]	TRADES [36]	Ours
NoATT*	7.594	7.673	7.736	7.768	7.748	7.744
FGSM [21]	13.924	12.996	9.119	9.140	9.149	9.202
Noise [19]	8.509	7.859	7.869	7.884	7.901	7.875
CW [32]	15.095	14.125	12.298	12.181	9.377	9.603
Universal [33]	14.927	13.834	9.903	9.797	9.644	9.711
PGD [31]	15.330	14.478	10.391	10.707	9.754	9.629
MADSA^{s}	15.582	14.387	10.566	10.748	9.795	9.633
MADSA^{+}	21.455	21.221	19.793	19.300	11.987	11.117
MADSA^{-}	18.894	18.728	16.816	16.817	11.497	10.865

* "NoATT" indicates that no attack is performed, and "NoDEF" indicates that no defense is applied. Therefore, the cell at the intersection of the NoATT row and the NoDEF column in the table corresponds to the STD result.

the proposed defense method achieved the best defense effect against $MADSA^{s}$, $MADSA^{-}$, and $MADSA^{+}$, with objectives of 9.633, 11.117, and 10.865, respectively.

In summary, the simulation results demonstrate that the proposed attack and defense methods still exhibit significant advantages over existing approaches in the case study based on a real-world ADN.

5 Conclusions and Discussions

MADRL-based optimal dispatching models for ADNs represent a future trend, revealing a significant gap in security research that needs to be addressed. This paper proposes a powerful attack algorithm for MADRL optimal dispatching of ADNs, capable of executing both single-agent and full-agent attacks. The algorithm can actively control the action direction of all attacked agents during full-agent attack, maximizing ADN deterioration. Additionally, a defense strategy is proposed to handle the integration of adversarial samples, with its feasibility theoretically proven. These insights address the gap in MADRL-based ADN dispatching and may provide forward-looking guidance for studies of adversarial attacks in other physical-cyber systems.

Although the proposed defense method MAGLD demonstrates promising performance in simulated environments, it may have some limitations when considered for long-term deployment. First, during extended operation in real-world systems, the environment may gradually change, requiring periodic fine-tuning of the defense policy. While such fine-tuning is generally lightweight and does not disrupt the core structure of the model, our current work does not explicitly evaluate long-term stability or online adaptation. Second, if the optimization objectives of the system evolve over time—such as changes in operational goals or constraints in future smart grids—the pre-trained model may become less effective. In such cases, adaptive mechanisms beyond manual tuning would be needed, which are not fully addressed in this paper. This study suggests two key future research directions: (1) for very long term extended operations, developing self-adaptive mechanisms to handle evolving grid optimization objectives without manual intervention and (2) research more covert partial-observation attacks that strategically alter only selected sensor data while preserving attack efficacy, along with corresponding enhancements to MAGLD's defense capabilities against such sophisticated threats.

A Proof of Lemma 3.1

For each agent i, after introducing the gradient leveling term, the optimization process not only minimizes the original loss function $\mathcal{L}(\theta)$ but also minimizes the squared ℓ_2-norm of the gradient. During the optimization process, the update rule is:

$$\nabla_\theta \mathcal{L}_{\text{smooth}}(\theta) = \nabla_\theta \mathcal{L}(\theta) + \frac{\lambda}{n} \sum_{i \in \mathcal{A}} \nabla_\theta \left(\frac{1}{2} \left\| \nabla_{\theta_i} \mathcal{L}_i(\theta_i) \right\|_2^2 \right). \tag{24}$$

By applying the chain rule, the gradient of the squared ℓ_2-norm term can be expressed as

$$\nabla_\theta \left(\frac{1}{2} \left\| \nabla_{\theta_i} \mathcal{L}_i(\theta_i) \right\|_2^2 \right) = \nabla_{\theta_i} \mathcal{L}_i(\theta_i)\, \nabla_{\theta_i}^2 \mathcal{L}_i(\theta_i), \tag{25}$$

which recovers the gradient–Hessian interaction term in (24).

According to Equation (18), since $\nabla_{\theta_i} \mathcal{L}_i(\theta_i)$ represents the loss gradient with respect to observation $\boldsymbol{o}_i$, the process of optimizing $\mathcal{L}_{\text{smooth}}(\theta)$ explicitly penalizes large gradient magnitudes. As a result, the optimization drives $\|\nabla_{\theta_i} \mathcal{L}_i(\theta_i)\|_2$ to become smaller, thereby achieving the purpose of gradient leveling.

B Proof of Lemma 3.2

Denote the loss function of agent i under observation $\boldsymbol{o}_i$ as $\mathcal{L}_i(\theta_i, \boldsymbol{o}_i)$. For a small perturbation $\boldsymbol{\eta}$ applied to the observation, the gradient of the loss with respect to θ_i can be approximated by a finite-difference scheme:

$$\nabla_{\theta_i} \mathcal{L}_i(\theta_i) = \frac{\partial \mathcal{L}_i(\theta_i, \boldsymbol{o}_i)}{\partial \theta_i} \approx \frac{\mathcal{L}_i(\theta_i, \boldsymbol{o}_i + \boldsymbol{\eta}) - \mathcal{L}_i(\theta_i, \boldsymbol{o}_i)}{\|\boldsymbol{\eta}\|_2}. \tag{26}$$

Equivalently, the above approximation can be expressed component-wise as

$$\left. \frac{\mathcal{L}_i(\theta_i, \boldsymbol{o}_i + \eta_h \boldsymbol{e}_h) - \mathcal{L}_i(\theta_i, \boldsymbol{o}_i)}{\eta_h} \right|_{h=1}^{|\boldsymbol{\eta}|}, \tag{27}$$

where η_h denotes the h-th element of $\boldsymbol{\eta}$ and $\boldsymbol{e}_h$ is the unit vector along the h-th dimension.

Let $\boldsymbol{o}_i' = \boldsymbol{o}_i + \boldsymbol{\delta}_i$ denote the adversarial observation. Then the induced loss difference is

$$\mathcal{L}_i(\theta_i, \boldsymbol{o}_i') - \mathcal{L}_i(\theta_i, \boldsymbol{o}_i) = \mathcal{L}_i(\theta_i, \boldsymbol{o}_i + \boldsymbol{\delta}_i) - \mathcal{L}_i(\theta_i, \boldsymbol{o}_i). \quad (28)$$

Since $\|\boldsymbol{\delta}_i\|_2 \leqslant \epsilon$ and $\epsilon \to 0$, the dependence of $\boldsymbol{\delta}_i$ on θ_i is negligible. By identifying $\boldsymbol{\delta}_i$ with $\boldsymbol{\eta}$ in (26) and (28), we obtain

$$\|\nabla_{\theta_i} \mathcal{L}_i(\theta_i, \boldsymbol{o}_i)\|_2 \approx \frac{|\mathcal{L}_i(\theta_i, \boldsymbol{o}_i') - \mathcal{L}_i(\theta_i, \boldsymbol{o}_i)|}{\|\boldsymbol{\delta}_i\|_2}. \quad (29)$$

Substituting (29) into the gradient-smoothing objective yields

$$\begin{aligned} \mathcal{L}_{\text{smooth}}(\theta) &= \mathcal{L}(\theta) + \frac{\lambda}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \|\nabla_{\theta_i} \mathcal{L}_i(\theta_i, \boldsymbol{o}_i)\|_2 \\ &= \mathcal{L}(\theta) + \lambda \mathbb{E}_{i \in \mathcal{A}} \left[\|\nabla_{\theta_i} \mathcal{L}_i(\theta_i, \boldsymbol{o}_i)\|_2 \right] \\ &\approx \mathcal{L}(\theta) + \lambda \mathbb{E}_{i \in \mathcal{A}} \left[\frac{|\mathcal{L}_i(\theta_i, \boldsymbol{o}_i') - \mathcal{L}_i(\theta_i, \boldsymbol{o}_i)|}{\|\boldsymbol{\delta}_i\|_2} \right]. \end{aligned} \quad (30)$$

Absorbing the constant factor $\|\boldsymbol{\delta}_i\|_2^{-1}$ into the smoothing coefficient yields an equivalent objective up to a scaling factor. Consequently, the above formulation admits the following Monte Carlo approximation:

$$\mathcal{L}_{\text{smooth}}(\theta) \approx \mathbb{E}_{i \in \mathcal{A}} \Big[\lambda \mathcal{L}_i(\theta_i, \boldsymbol{o}_i') + (1-\lambda) \mathcal{L}_i(\theta_i, \boldsymbol{o}_i) \Big]. \quad (31)$$

Therefore, adversarial training with a weighted mixture of perturbed and clean samples can be interpreted as a Monte Carlo approximation of a gradient-smoothing regularization, which completes the proof of Lemma 3.2.

C BDD based stealthiness evaluation of MADSA samples

The stealthiness of the proposed attack is evaluated using the χ^2-based BDD test with a significance level $\alpha = 0.05$, which corresponds to a detection threshold $\tau = 49.8$. To simulate a realistic monitoring environment, Gaussian white noise $\mathcal{N}(0, 0.02^2)$. is added to the power measurements, and the empirical evasion rate is calculated through 1,000 Monte Carlo trials for each attack magnitude $\epsilon \in \{0, 0.05, 0.10, 0.20\}$.

Table 8 Result of BDD based stealthiness evaluation

	Average Residual $J(\hat{\mathbf{x}})$	Evasion Rate
$\epsilon = 0$	35.2	98.0%
$\epsilon = 0.05$	44.5	89.2%
$\epsilon = 0.1$	56.4	41.3%
$\epsilon = 0.2$	132.7	5.8%

The results show that for $\epsilon = 0.05$, the average residual $J(\hat{\mathbf{x}}) = 44.5 \leqslant \tau\ (49.8)$. This leads to a high evasion rate of 89.2%, proving that the attack is sufficiently stealthy to bypass BDD. Given that the previous experiments have already demonstrated the effectiveness of MADSA with $\epsilon = 0.05$, this magnitude is considered optimal for balancing attack impact and stealthiness.

Reference

[1] X. Jia, Y. Xia, Z. Yan, H. Gao, D. Qiu, J. M. Guerrero, and Z. Li, Coordinated operation of multi-energy microgrids considering green hydrogen and congestion management via a safe policy learning approach, *Applied Energy*, vol. 401, p. 126611, 2025.

[2] M. Thirunavukkarasan, K. P. Remamany, M. P. Vaishnnave, S. A. S. A. Mary, G. G. Devarajan, and R. P. Mahapatra, Intelligent retrieval and secure content generation in consumer healthcare electronics using quantum blockchain and edge-fog-cloud intelligence, *IEEE Transactions on Consumer Electronics*, pp. 1–8, 2025, early Access.

[3] Y. Su, M. Tan, and J. Teh, Short-term transmission capacity prediction of hybrid renewable energy systems considering dynamic line rating based on data-driven model, *IEEE Transactions on Industry Applications*, pp. 1–11, 2025.

[4] D. Silver *et al.*, Mastering the game of go with deep neural networks and tree search, *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[5] Z. Xie and P. Dames, Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles, *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 2700–2719, 2023.

[6] R. Zhu *et al.*, Adaptive broad-deep reinforcement learning for intelligent traffic light control, *IEEE IoT J.*, 2024, early Access.

[7] Y. Lu *et al.*, Deep reinforcement learning based optimal scheduling of active distribution system considering distributed generation, energy storage and flexible load, *Energy*, vol. 271, p. 127087, 2023.

[8] H. Gao *et al.*, A two-stage multi-agent deep reinforcement learning method for urban distribution network reconfiguration considering switch contribution, *IEEE Trans. Power Syst.*, 2024, early Access.

[9] M. Hosseini, L. Rodriguez-Garcia, and M. Parvania, Hierarchical combination of deep reinforcement learning and quadratic programming for distribution system restoration, *IEEE Trans. Sustain. Energy*, vol. 14, no. 2, pp. 1088–1098, 2023.

[10] Y. Xiang, Y. Lu, and J. Liu, Deep reinforcement learning based topology-aware voltage regulation of distribution networks with distributed energy storage, *Appl. Energy*, vol. 332, p. 120510, 2023.

[11] E. Samadi, A. Badri, and R. Ebrahimpour, Decentralized multi-agent based energy management of microgrid using reinforcement learning, *Int. J. Electr. Power Energy Syst.*, vol. 122, p. 106211, 2020.

[12] D. Cao *et al.*, Decentralized graphical-representation-enabled multi-agent deep reinforcement learning for robust control of cyber-physical systems, *IEEE Trans. Rel.*, 2024, early Access.

[13] F. Hu, Y. Deng, and A. H. Aghvami, Scalable multi-agent reinforcement learning for dynamic coordinated multipoint clustering, *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 101–114, 2022.

[14] P. Chen *et al.*, Multi-agent reinforcement learning for decentralized resilient secondary control of energy storage systems against dos attacks, *IEEE Trans. Smart Grid*, vol. 13, no. 3, pp. 1739–1750, 2022.

[15] D. Chen *et al.*, Powernet: Multi-agent deep reinforcement learning for scalable power grid control, *IEEE Trans. Power Syst.*, vol. 37, no. 2, pp. 1007–1017, 2021.

[16] B. Zhang *et al.*, Multi-agent deep reinforcement learning based distributed control architecture for interconnected multi-energy microgrid energy management and optimization, *Energy Convers. Manag.*, vol. 277, p. 116647, 2023.

[17] Y. Chen, H. Ma, Y. Wang, and X. Zhang, A novel dynamic monitoring method based on co-integration analysis for non-stationary processes, *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–14, 2025.

[18] A. Selim *et al.*, Adaptive deep reinforcement learning algorithm for distribution system cyber attack defense with high penetration of ders, *IEEE Trans Smart Grid*, vol. 15, no. 4, pp. 4077–4089, 2024.

[19] L. Zeng *et al.*, Physics-constrained vulnerability assessment of deep reinforcement learning-based scopf, *IEEE Trans. Power Syst*, vol. 38, no. 3, pp. 2690–2704, 2022.

[20] Z. Peng, Q. Yang, D. Li, F. Zhang, and P. Song, Adversarial attacks on deep reinforcement learning applications in electric vehicle charging scheduling: A dual-stage attack framework, *Applied Soft Computing*, vol. 181, p. 113450, 2025.

[21] Y. Zheng *et al.*, Vulnerability assessment of deep reinforcement learning models for power system topology optimization, *IEEE Trans Smart Grid*, vol. 12, no. 4, pp. 3613–3623, 2021.

[22] P. Chen *et al.*, Dynamic event-triggered output feedback control for load frequency control in power systems with multiple cyber attacks, *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 52, no. 10, pp. 6246–6258, 2022.

[23] X. Chen *et al.*, Advdiffuser: Natural adversarial example synthesis with diffusion models, in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023.

[24] L. Ye, K. Zhang, B. Jiang, and S. Simani, Enhancing aerospace fault diagnosis with conditioned multiscale generative adversarial networks, *IEEE Transactions on Cybernetics*, vol. 56, no. 2, pp. 816–827, 2026.

[25] J. Hao *et al.*, Exploration in deep reinforcement learning: From single-agent to multiagent domain, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8762–8782, 2024.

[26] Z. Dai, M. Tan, Y. Yang, X. Liu, Y. Su, and K. Li, Towards adversarial robustness in madrl-based ev charging scheduling: A tailored multi-criteria adversarial attack and defense framework, *IEEE Transactions on Transportation Electrification*, pp. 1–16, 2026, early Access.

[27] F. Santoso and A. Finn, A data-driven cyber–physical system using deep-learning convolutional neural networks: Study on false-data injection attacks in an unmanned ground vehicle under fault-tolerant conditions, *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 53, no. 1, pp. 346–356, 2022.

[28] Y. Su *et al.*, Optimal dispatching for ac/dc hybrid distribution systems with electric vehicles: Application of cloud-edge-device cooperation, *IEEE Trans Intell Transp Syst*, vol. 25, no. 3, pp. 3128–3139, 2024.

[29] R. Rigo-Mariani and V. Vai, An iterative linear distflow for dynamic optimization in distributed generation planning studies, *International Journal of Electrical Power & Energy Systems*, vol. 138, p. 107936, 2022.

[30] A. Wagenmaker, K. Huang, L. Ke, K. Jamieson, and A. Gupta, Overcoming the sim-to-real gap: Leveraging simulation to learn to explore for real-world rl, in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 78 715–78 765.

[31] R. B. Lanfredi, J. D. Schroeder, and T. Tasdizen, Quantifying the preferential direction of the model gradient in adversarial training with projected gradient descent, *Pattern recognition*, vol. 139, p. 109430, 2023.

[32] F. S. Atedjio, J.-P. Lienou, F. F. Nelson, S. S. Shetty, and C. A. Kamhoua, A defensive strategy against android adversarial malware attacks, *IEEE Access*, vol. 12, pp. 169 432–169 441, 2024.

[33] L. Schwinn, D. Dobre, S. Xhonneux, G. Gidel, and S. Gunnemann, Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space, in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 9086–9116.

[34] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, Towards deep learning models resistant to adversarial attacks, *arXiv preprint arXiv:1706.06083*, 2017.

[35] E. Wong, L. Rice, and J. Z. Kolter, Fast is better than free: Revisiting adversarial training, in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2020, pp. 1–12.

[36] C. P. Lau, J. Liu, H. Souri, W.-A. Lin, S. Feizi, and R. Chellappa, Interpolated joint space adversarial training for robust and generalizable defenses, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13 054–13 067, 2023.

[37] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, and C.-J. Hsieh, Robust deep reinforcement learning against adversarial perturbations on state observations, *Advances in neural information processing systems*, vol. 33, pp. 21 024–21 037, 2020.

[38] H. Zhang, H. Chen, D. S. Boning, and C.-J. Hsieh, Robust reinforcement learning on state observations with learned optimal adversary, in *International Conference on Learning Representations*, 2021.

[39] X. Li *et al.*, Multi-agent drl for resource allocation and cache design in terrestrial-satellite networks, *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5031–5042, 2022.

[40] H. Kang *et al.*, Cooperative uav resource allocation and task offloading in hierarchical aerial computing systems: A mappo-based approach, *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10 497–10 509, 2023.

[41] M. Jadoon *et al.*, Learning random access schemes for massive machine-type communication with marl, *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 2, pp. 95–109, 2023.