A Robust Power Grid Scheduling Method Based on Improved Transformer and Reinforcement Learning

< Wenjiang Wei>[1], <Xiaoxin Gao>[2], <Ming Sun>[3]

[1] North China Electric Power University, School of Control and Computer Engineering,
Beinong Road, Beijing, China
wwj19990725@163.com

[2] Beijing China-Power Information Technology Co., Ltd.,
Beiqing Road, Beijing, China
13521395850@163.com

[3] Beijing China-Power Information Technology Co., Ltd.,
Beiqing Road, Beijing, China
13501225160@139.com

Abstract: ***The integration of renewable energy with explosive growth in scale, inherently intermittent and stochastic, poses severe challenges to grid dispatch, while contemporary artificial intelligence technologies, rising in parallel with computational power, may serve as potent tools to address complex grid control issues. Particularly, reinforcement learning-based methods have been widely utilized and have achieved certain successes in both theoretical research and practical applications in grid dispatch. However, deploying reinforcement learning models into real-world grid dispatch tasks requires ensuring their robustness against sudden disturbances, an area that requires further investigation. In this paper, we propose a method of adversarial training that integrates historical information encoding. Specifically, we employ adversarial Markov policies to learn attack strategies, then utilize adversarial training methods to enhance the model's robustness against adversary attacks. Building upon this, we utilize GTrXL (a variant of Transformer) to encode current and historical state information, enabling the model to make more robust decisions over longer observation horizons. We experimented with the proposed method in the IEEE-14 environment provided by the L2RPN competition and compared it with the algorithms of the competition's award-winning participants, verifying the effectiveness of our approach.***

Keywords: Grid dispatch, Reinforcement learning, Adversarial training, Historical information encoding**.**

1. Introduction

Grid dispatch is the process of monitoring, controlling, and optimizing the operation of the power system to ensure the safety, stability, and cost-effectiveness of electricity supply. With the widespread integration of renewable energy into the grid and the surge in residential electricity demand, ensuring the secure operation of the power system has become a significant challenge. From the perspective of energy supply, renewable energy generation is affected by factors such as weather and environment, exhibiting obvious randomness, intermittency, and low dispatchability[1]. From the perspective of energy demand, electric vehicles and other devices with distinct charging demand characteristics further exacerbate the uncertainty and variability of the grid[2]. Traditional manual scheduling methods are no longer sufficient to meet the needs of grid dispatch under dynamic changes in supply and fluctuating demand conditions, hence there is an expectation to seek a more robust grid dispatch method.

Some researchers have attempted to optimize grid dispatch from the perspective of network structure. Baranwal proposed a distributed control architecture for coordinating the operation of multiple DC-DC converters in DC microgrids to achieve stable voltage regulation and power sharing[3]. Mohsen Hamzeh introduced a novel method that selects the optimal configuration of power network systems adapted to network protocols through graph theory and education-based optimization algorithms to reduce expected energy non-supply and achieve significant reliability improvements in practical cases[4]. However, these hardware modifications are not only resource-intensive but also have certain limitations for specific issues. Conversely, model-based approaches transform grid control problems into constraint problems[5]-[6], offering a solid theoretical foundation. They consume fewer resources compared to network structure modifications and are easier to scale. However, the increasingly complex grid structures and uncertainties in grid operations make it impossible to accurately describe and construct highly nonlinear, complex system models using existing mathematical tools.

In recent years, with the rapid development of artificial intelligence technology, reinforcement learning (RL) as one of its branches has made significant achievements in complex control fields[7]-[9]. The AlphaGo developed by the DeepMind team based on deep reinforcement learning (DRL) technology defeated the world champion in the 2016 Go competition[7]. According to researchers' estimates, the number of legal Go game positions far exceeds the number of atoms in the observable universe, demonstrating the immense complexity and variability of Go. On other fronts, the Google X team utilized DRL to train robotic arms for automatic door opening and item retrieval[8], while the Uber team used DRL to train game characters to drive vehicles in real environments[9]. These studies illustrate the significant advantages of reinforcement learning in solving complex control problems, prompting some researchers to attempt its application in the field of grid dispatch, including photovoltaic and energy storage control[10], voltage and current control[11]-[12], and grid topology control[13]-[14], among others. For more examples of reinforcement learning applications in power systems, refer to literature[15].

The above-mentioned studies demonstrate the excellent performance of reinforcement learning algorithms in the field of grid dispatch. However, as grid dispatch gradually transitions towards intelligence, it also brings certain risks—smart grids may face threats of malicious attacks. For example, in 2015, the information system of a Ukrainian energy distribution company was hacked, leading to a power outage lasting up to 6 hours and affecting power supply services for over 230,000 people[16]. On the other hand, studies have shown that reinforcement learning is susceptible to noise interference. Even a minor disturbance in the state space can lead to suboptimal actions from a fully optimized reinforcement learning agent[17]. As the power grid system is a critical infrastructure affecting people's livelihoods, researching robust dispatch methods under uncertain disturbances has become a top priority in current grid optimization and dispatch tasks.

In the field of reinforcement learning, researchers have classified uncertain attacks on reinforcement learning into four categories based on their components[18], with the primary focus being on adding adversarial perturbations to the state space. A small portion involves perturbing the reward function and state space, and different defense methods are employed against different types of attacks. Kos and Song used random noise and FGSM to generate adversarial inputs to train their models[19], demonstrating their algorithm's resistance to attacks of the same type. Pinto proposed Robust Adversarial Reinforcement Learning (RARL) as a method for robust policy learning in the presence of adversaries, where adversaries have specifically set rewards aimed at finding the state trajectory with the lowest reward[20]. Wang et al. pointed out that reward functions are susceptible to three types of noise[21]: intrinsic noise, application-specific noise, and adversarial noise, and they proposed a reward confusion matrix to generate rewards. Smirnova et al. proposed a distributed robust policy generation with dynamic risk criteria[22], which can prevent agents from taking suboptimal actions based on the risk criteria.

In the field of power systems, researchers are also dedicated to studying robust grid dispatch. The L2RPN grid dispatch competition, jointly organized by the French grid operator and the Electric Power Research Institute, presented two tracks in 2020. One track aimed to increase the time of stable grid operation as much as possible in the presence of external attacks and uncertainties. The champion team, Baidu, utilized a search-based planning algorithm to filter out illegal actions by searching the state space in advance[23], ensuring that the actions taken strictly adhere to the constraints. The runner-up team proposed a framework of "teacher-expert-lower-grade student-upper-grade student". The teacher and expert used a greedy strategy to enumerate all possible dispatch operations and filter out some low-frequency operations to narrow the action space. The lower-grade student learned by imitation to mimic the expert's strategy, while the upper-grade student learned from the lower-grade student's Critic network but focused more on long-term rewards, thus achieving better performance. Apart from the competition participants, Zeng et al. focused on research on false data injection that can bypass adverse data monitoring mechanisms in power systems[24]. Xu proposed a graph attention-based reinforcement learning method to achieve robust active power correction control[25], while Pan proposed using adversary agents for adversarial training to enhance the robustness of grid dispatch reinforcement learning models[26]. They conducted experiments on multiple award-winning agents and improved their robustness against disturbances.

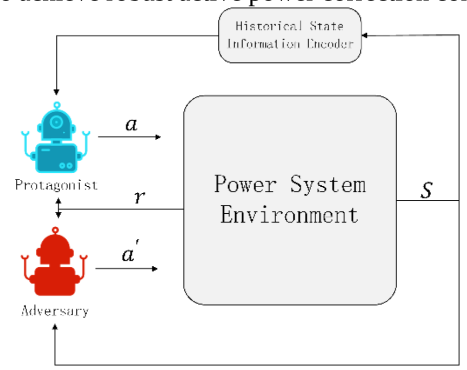

Figure 1 Adversary training frame

Our work, similar to Pan's, primarily focuses on studying the robustness of reinforcement learning agents in grid scheduling against unknown disturbances. We define disturbances such as natural disasters, human interventions, and unknown network attacks as adversary agents with properties similar to the protagonist agent but with opposite optimization objectives. Their goal is to minimize the accumulated rewards in the environment where the protagonist exists. We then employ adversarial training between the adversary and protagonist agents, where the protagonist aims to maximize accumulated rewards in the presence of the adversary, ultimately obtaining a robust model capable of effectively handling adversary disturbances.

To further enhance the model's robustness, inspired by the work of Zhang et al[27], we attempt to break free from the Markovian constraints by utilizing longer historical information for grid scheduling decisions, as shown in Figure 1. Previous research by Espehol et al[28]. utilized RNN and LSTM as memory units to provide historical state information to agents. However, inherent issues such as gradient vanishing, exploding, and difficulty in capturing long-range dependencies hindered their extensive application in the reinforcement learning domain.

Recent related works empirically demonstrate that self-attention architectures, such as Transformers, outperform traditional recursive architectures (e.g., LSTM) in various fields like language modeling[29], machine translation [30], showcasing their strong performance. This has spurred scholars to delve deeper into the integration of Transformers with reinforcement learning[31]. However, Parisotto pointed out that standard Transformer architectures are challenging to optimize and perform poorly even with complex training techniques when applied to reinforcement learning [32]. To address this, an improved Transformer architecture GTrXL was proposed, enhancing stability and convergence speed by modifying the original Transformer structure and introducing gating mechanisms.We utilize GTrXL as the historical state encoder to merge current and historical states, providing them to the agent as scheduling decision bases, thereby aiding in making more robust decisions. Additionally, inspired by Pan's training techniques, we further discuss and experimentally verify the influence of pre-trained models and different adversary action spaces on adversarial training results.

In summary, this research makes the following original contributions:

(1)Building upon the use of reinforcement learning to achieve fundamental grid scheduling objectives, we further enhance the model's robustness against uncertain disturbances through adversarial training techniques.

(2) By introducing the GTrXL module, we extract efficient and robust state representations from historical state information, further enhancing the robustness of the model.

2. Grid Operation Model

In this section, we have introduced the components of grid topology and the basic objectives of grid dispatch. In the next section, we will explain how to integrate grid dispatch tasks into a reinforcement learning framework and perform robust optimization.

2.1 The composition of grid topology

The topology of the power grid can be abstracted as a graph structure$G = \{V, E\}$, where E represents the set of power lines used for electricity transmission, and V represents the set of substations. Typically, a substation connects to either a power generator or a load on one end, and to other substations on the other end (although there are cases where substations are connected on both ends). Internally, substations generally have a dual-bus structure. Changing the connection of power lines on the bus can alleviate the overload situation of a particular line. As shown in the figure 2, the current of power line y_4 is 118, exceeding its thermal limit of 100. By modifying the bus S_4, the current of y_4 is restored to its thermal limit. Additionally, optimization of the load on each line in the power grid can be achieved by adjusting the power output of generators and loads (referred to as redispatch).

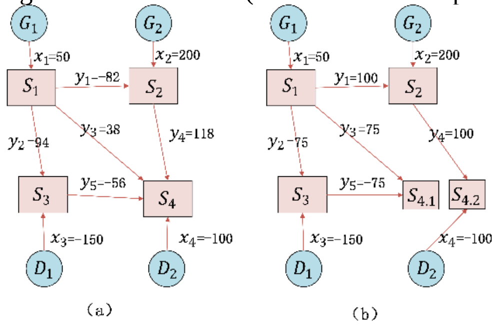

Figure 2 Redispatch

2.2 Basic Objectives of Grid Dispatch

In power system control tasks, in addition to ensuring the stable operation of the grid, operators primarily focus on minimizing operational costs over a certain period of time. Specifically, within a time period T, each scheduling operation $a_t$executed at time step t incurs a corresponding cost consumption C, as shown in Equation 1.

$$\min_{a_t} \quad C(s_t, a_t) \tag{1}$$

$$\text{s.t.} \quad f_{ga}(a_t) = 0 \tag{1a}$$

$$f_{ha}(a_t) \leqslant 0 \tag{1b}$$

$$f_{gs}(s_t) = 0 \tag{1c}$$

$$f_{hs}(s_t) \leqslant 0 \tag{1d}$$

Equation 1 represents the total dispatch cost C over a period of time T, including topological changes and power losses, among others. Our objective is to find a sequence of actions $[a_0, a_1, \dots, a_T]$ that minimizes the total cost. Each scheduling action must satisfy certain constraints: Equation (1a) (1c) represents node power balance and current equations, while Equation (1b)(1d) represents hardware system constraints such as line thermal limits and node voltage limits. Although solving this optimization problem using mathematical programming can be a solution, the increasingly complex scale of the power grid limits the applicability of this method. Therefore, we employ a more powerful tool in addressing complex control problems reinforcement learning to solve this optimization problem.

3. Robust Grid Dispatch Design

In this section, we first define the grid dispatch model as a Markov Decision Process and solve it using reinforcement learning. Then, we enhance the robustness of the reinforcement learning model through adversarial training, specifically training adversary agents and protagonist agents. Finally, we further enhance the robustness of the model by incorporating a historical state encoder.

3.1 Reinforcement Learning-based Grid Dispatch Model

In the framework of reinforcement learning, we define the grid dispatch model as a Markov Decision Process, which can be represented as a 4-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R)$. At time step t, the agent receives a state vector $s_t \in \mathcal{S}$ composed of information data from the grid, including network topology information, active power of generators and loads, reactive power, current flow, and line thermal limits. Then, it makes a scheduling decision $a_t \in \mathcal{A}$, such as topological modifications and redispatching mentioned in the previous section. The grid environment generates the next state s_{t+1} based on the current state s_t, scheduling decision at, and transition probability $\mathcal{P}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$. The transition probability $\mathcal{P}$ is provided by the grid environment simulator through power flow calculation. At the same time, the agent receives a reward R, which is negatively correlated with the scheduling cost at this time step. If a power outage occurs, the minimum reward is obtained (usually set to 0, while the maximum reward is 1).

The objective of the reinforcement learning agent is to find an optimal policy π that maximizes the cumulative reward.

$$\max_{\theta} \quad J(\theta) = E_{\mathcal{P},\pi}\left[\sum_{t=1}^{T} R(s_t, a_t)\right] \tag{2}$$

$$\text{s.t.} \quad a_t = \pi_\theta(s_t)$$

$$a \in \mathcal{A}, s \in S$$

Where $J(\theta)$ represents the expected cumulative reward under the current policy. By referencing the choices of other participants in the competition and comparing the effectiveness of their algorithms, we ultimately chose the Proximal Policy Optimization (PPO) method based on policy gradients to solve the optimization problem described by the equation above.

3.2 Adversarial Training in Reinforcement Learning

An adversarial grid dispatch environment can be represented as a two-player Markov game process, where the Markov Decision Process of this game can be expressed as a tuple $(\mathcal{S}, \mathcal{A}1, \mathcal{A}2, \mathcal{P}, R)$. Here, $\mathcal{A}1$ and $\mathcal{A}2$ represent the action spaces of the protagonist agent and the adversary agent, respectively. It is important to note that we impose certain constraints on the adversary's action space (specific reasons and details will be explained in subsequent experiments). $\mathcal{A}2$ includes only a subset of operable lines. $\mathcal{P}: \mathcal{S} \times \mathcal{A}1 \times \mathcal{A}2 \times \mathcal{S} \rightarrow \mathbb{R}$ represents the transition probability, and $R: \mathcal{S} \times \mathcal{A}1 \times \mathcal{A}2 \rightarrow \mathbb{R}$ denotes the rewards for both agents.

Training of the Adversary Agent: For a protagonist agent's policy π_{θ}^{pro}, we aim to learn an adversary agent's policy $\pi_{\theta'}^{adv}$, whose objective is to alter the normal grid topology to induce an unsafe state in the grid system. In this state, the protagonist agent lacks experience in handling such situations and struggles to make appropriate scheduling decisions, resulting in power outages. The protagonist agent receives fewer cumulative rewards under the influence of the adversary.

$$\begin{aligned} &\underset{\theta'}{min} && J(\theta, \theta') \\ &\text{s.t.} && s_{t+1} \sim \mathcal{P}(s_{t+1} \mid s_t, a_t, a'_t) \\ & && a'_t = \pi_{\theta'}^{adv}(s_t), a_t = \pi_{\theta}^{pro}(s_t) \text{ fixed } \theta \\ & && a' \in \mathcal{A}2, a \in \mathcal{A}1, s \in \mathcal{S} \end{aligned} \tag{3}$$

With reference to Pan's adversary design, all adversary attack methods in this paper are black-box attacks, meaning the adversary does not have knowledge of the policy of the protagonist agent being attacked. Specifically, we use adversary adv_x for adversarial training to obtain a robust protagonist pro_x, and then use adversary adv_y which has no training interaction with pro_x to test the robustness of pro_x. This design reflects the reality where adversaries often cannot obtain the true scheduling policy.

Training of the Robust Protagonist Agent: We have obtained an adversary agent with significant perturbation capabilities using the above method, which represents unknown disturbances in reality, whether natural or human-made. We aim to enhance the protagonist agent's robustness to unknown disturbances by allowing it to interact with adversarial scenarios through adversarial training. The optimization problem for the robust agent can be represented by equation 4:

$$\max_{\theta} \; J(\theta, \theta') \tag{4}$$

The process of interaction between the protagonist and adversary agents in the environment and collecting trajectories is as follows:

1. The adversary agent observes the state s_{t-1} and takes an interference action a'. The environment updates to state s_t based on the transition probability $\mathcal{P}$.
2. The protagonist agent observes the state s_t and takes a scheduling action a_t, receiving feedback reward r_t. The environment updates to state s_{t+1} based on the transition probability $\mathcal{P}$.
3. Collecting trajectory information (s_t, a_t, r_t, s_{t+1}).

3.3 Historical State Information Encoder

To further enhance the model's robustness, we attempt to overcome the limitations of Markovianity and utilize GTrXL (a variant of Transformer) as the encoder for historical state information of the grid scheduling agent. The structure of a single module is depicted in Figure 3 (right). Compared to the original Transformer structure shown in Figure 3 (left), GTrXL alters the sequence of Layer-Norm in the module. This adjustment enables it to achieve an identity mapping from the input of the first layer module to the output of the last layer module, allowing the encoded state to contain more information about the current moment, aiding the model in learning Markovian policies before the attention mechanism is fully optimized during the initial training stages. Additionally, GTrXL introduces a Gating Layer, which controls information flow using gating mechanisms, further enhancing the model's performance and stability. The GTrXL single-layer module can be described by the following equation 5:

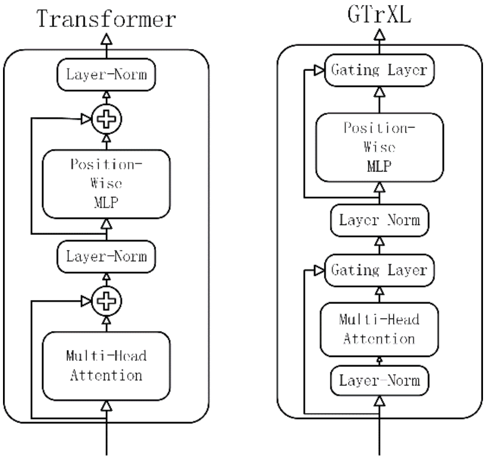

Figure 3 Compare Transformer and GTrXL

$$\begin{aligned} \overline{Y}^{(l)} &= \mathrm{RMHA}\left(LN\left(\left[SG\left(M^{(l-1)}\right), E^{(l-1)}\right]\right)\right) \\ Y^{(l)} &= g_{MHA}^{(l)}\left(E^{(l-1)}, \mathrm{Re}\,LU\left(\overline{Y}^{(l)}\right)\right) \\ \overline{E}^{(l)} &= f^{(l)}\left(LN\left(Y^{(l)}\right)\right) \\ E^{(l)} &= g_{MLP}^{(l)}\left(Y^{(l)}, \mathrm{Re}\,LU\left(\overline{E}^{(l)}\right)\right) \end{aligned} \tag{5}$$

Where $l \in [0, L]$ represents the index of the module layer, E represents the input of the current layer, which is also the output of the previous layer. $M(l) \in \mathbb{R}^{T \times D}$ represents a tensor used to store historical state information, and SG denotes the stop-gradient function. LN and $RMHA$ denote layer normalization and multi-head attention layers, respectively. g_{MHA} and g_{MLP} represent two gated recurrent units, and f denotes a multi-layer perceptron (MLP).

4. Experiment

Finally, we validated the effectiveness of our proposed method through experiments. We first describe our experimental setup. Then, through comparative experiments, we demonstrate the effectiveness of our algorithm. Additionally, through ablation experiments, we illustrate the impact of historical information encoding and pre-training models on adversarial training results.

4.1 Experimental Setup

Experimental Environment:We utilized the IEEE-14 power grid as the experimental environment. This power grid environment comprises multiple scenarios, with each scenario specifying parameter variations for power plants and loads during simulation. Each scenario has a duration of 864 time steps, with each time step approximately equivalent to 5 minutes, corresponding to approximately 3 days of real-world power grid operation for each scenario.

Baselines:To validate the robustness of our reinforcement learning agent against adversarial attacks, we compared its performance with publicly available code from L2RPN competition winners:

1. PPO: Baseline PPO algorithm provided by RET-France, the official organizer of the L2RPN competition. Many participants in past competitions have used this algorithm as a baseline or built upon it for improvements.
2. KAIST: Proposed a hierarchical policy architecture with posterior state representations, winning the L2RPN WCCI 2020 competition.
3. NANYANG: Utilized two agents for grid scheduling, each employing different strategies, techniques, and trained on random datasets. Achieved third place in the L2RPN WCCI 2020 competition.

Policy Evaluation: To verify the robustness of the algorithm, we referred to the adversary settings outlined in Pan et al. Specifically, they are as follows:

1. No attack: The adversary does not launch any attacks on the power grid.
2. Random attack: The adversary randomly disrupts a power line in the network when attacking the power grid.
3. Learned attack:The adversary, obtained through adversarial training, is capable of identifying vulnerabilities in the current state of the power grid and targeting critical lines for disruption.

we observed that disconnecting certain lines could immediately cause a blackout in the power grid. Therefore, we only allowed a subset of lines to be attacked, reflecting the limited strength of adversaries in reality. During the testing phase, to prevent the power grid environment from collapsing too quickly or prematurely, we configured the adversary to attack once every 20 steps but not immediately upon activation. Instead, the adversary begins attacking only after the protagonist agent has been running for more than 20 steps.

Parameter Settings:The historical state encoding network encodes the input state into a vector of size 400. The encoded state is concatenated with historical information of size 864 and fed into a single-layer GTrXL module. The attention layer is a single-headed attention layer, and the output dimension of the module is a vector of size 400, which then enters the Actor-Critic network. The shared layer has 200 neurons, and the number of neurons in both the Actor and Critic networks is [200, 200].

For each model, we trained for a total of 2000 epochs. During each epoch, we collected data for 2000 time steps and updated the network using this data 10 times. Additionally, the discount factor γ for the model was set to 0.99, and the clipping parameter was set to 0.2.

4.2 Performance Comparison

We conducted comparative experiments to validate that our proposed algorithm enhances the robustness of reinforcement learning in grid scheduling. In the IEEE-14 power grid environment, we compared the performance of the standard PPO, KAIST, NANYANG, and our algorithm under different adversary scenarios. As shown in the Table 1, we first horizontally compared the performance of adversaries. Taking the PPO algorithm as an example, in an environment without adversaries, the algorithm could run for over 600 time steps. However, after introducing a random adversary, this number dropped to 45 steps. When replacing the random adversary with the learned adversary, the system couldn't withstand even a single attack, with the power grid collapsing immediately within 5 time steps after the adversary's attack. This demonstrates that well-trained agents are highly vulnerable when facing adversary attacks. Additionally, it indicates that we have learned a relatively stronger adversary, and engaging in adversarial training against stronger adversaries can yield more robust protagonist agents【26】.Vertically, our method showed a slight improvement compared to the standard PPO algorithm in the absence of disturbances, indicating that adversarial training can even enhance the model's performance in ordinary undisturbed environments. However, in environments with attacks, only our algorithm demonstrated some resistance against both random and learned adversaries, while the other three agents were quickly defeated by the adversaries. This highlights the robustness of our algorithm against disturbances from unknown adversaries.

Table 1: Performance comparison

		No attack	Random	Learned
PPO	reward	502.15±180.45	35.70±17.47	16.94±0.44
	step	619.2±226.14	45.1±20.77	21±0
Kaist	reward	**690.22±78.14**	**114.56±27.23**	35.12±14.23
	step	**827.6±104.3**	**147.32±35.27**	48.66±18.83
NAN-YANG	reward	610.32±150.74	68.36±23.39	34.27±11.28
	step	746.25±176.92	85.42±28.65	46.78±18.64
Our	reward	633.58±107.84	68.18±43.92	**120.18±74.4**
	step	787.4±133.95	85.1±53.76	**152.8±93.6**

4.3 Ablation Experiment

To validate the effectiveness of encoding historical state information, we conducted ablation experiments comparing the performance changes of models before and after the addition of historical state information encoding, both with and without adversarial training. Table 2 presents our comparative results.

Table 2: Historical State Information Encoder analyze

		No attack	Random	Learned
PPO	reward	502.15±180.45	35.70±17.47	16.94±0.44
	step	619.2±226.14	45.1±20.77	21±0
TPPO	reward	**656.30±115.15**	38.89±28.80	22.56±16.03
	step	**793.0±139.35**	50.9±34.22	29.6±19.46
Adv-PPO	reward	560.63±132.18	45.83±22.54	93.91±68.27
	step	724.0±170.22	60.3±28.46	120.0±86.53
Adv-TPPO	reward	633.58±107.84	**68.18±43.92**	**120.18±74.4**
	step	787.4±133.95	**85.1±53.76**	**152.8±93.6**

We found that in the absence of adversarial training, TPPO with added historical state encoding outperforms standard PPO when there are no adversaries present. This indicates that longer state sequences contribute to obtaining a stronger policy model. Additionally, from the training process graphs of both models (Figure 3), it can be observed that TPPO not only converges to a higher value but also exhibits smaller fluctuations after convergence, demonstrating the stabilizing effect of our state encoder during training.On the other hand, after adversarial training, the adv PPO algorithm shows a significant improvement when facing adversary attacks compared to PPO without adversarial training. It also demonstrates some improvement against random attacks, highlighting the effectiveness of our adversarial training. Incorporating historical state encoding into adv TPPO further enhances its performance, emphasizing the beneficial effect of historical state encoding in improving model robustness.

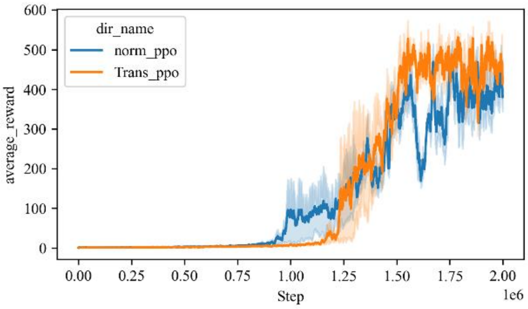

Figure 4 Validity of GTrXL module

5. Conclusion

In this work, we investigated the robust grid scheduling problem using reinforcement learning. We employed adversarial training to obtain powerful perturbation adversaries and further developed protagonist policies robust to adversary disturbances. Building upon this, we extended the observation range of protagonist policies by incorporating a historical state information encoding module, aiding in making more robust decisions. Our proposed approach was compared with algorithms from competition-winning participants in the IEEE-14 environment provided by the L2RPN competition, validating the robustness of our method against disturbances from unknown adversaries.

In the future, we plan to investigate the impact of minor perturbations in the state space on reinforcement learning models and integrate this into the framework presented in this paper.

References

[1] Ma H, Oxley L, Gibson J, et al. A survey of China's renewable energy economy[J]. Renewable and Sustainable Energy Reviews, 2010, 14(1): 438-445.

[2] Morrissey P, Weldon P, O’Mahony M. Future standard and fast charging infrastructure planning: An analysis of electric vehicle charging behaviour[J]. Energy policy, 2016, 89: 257-270.

[3] Baranwal M, Askarian A, Salapaka S, et al. A distributed architecture for robust and optimal control of DC microgrids[J]. IEEE Transactions on Industrial Electronics, 2018, 66(4): 3082-3092.

[4] Hamzeh M, Vahidi B, Nematollahi A F. Optimizing configuration of cyber network considering graph theory structure and teaching–learning-based optimization (GT-TLBO)[J]. IEEE Transactions on Industrial Informatics, 2018, 15(4): 2083-2090.

[5] Li Y, Huang R, Ma L. False data injection attack and defense method on load frequency control[J]. IEEE Internet of Things Journal, 2020, 8(4): 2910-2919.

[6] Zhang Y, Gatsis N, Giannakis G B. Robust energy management for microgrids with high-penetration renewables[J]. IEEE transactions on sustainable energy, 2013, 4(4): 944-953.

[7] Wang F Y, Zhang J J, Zheng X, et al. Where does AlphaGo go: From church-turing thesis to AlphaGo thesis and beyond[J]. IEEE/CAA Journal of Automatica Sinica, 2016, 3(2): 113-120.

[8] Xiao T, Jang E, Kalashnikov D, et al. Thinking while moving: Deep reinforcement learning with concurrent control[J]. arXiv preprint arXiv:2004.06089, 2020.

[9] Jha S S, Cheng S F, Lowalekar M, et al. Upping the game of taxi driving in the age of Uber[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32(1).

[10] Kofinas P, Doltsinis S, Dounis A I, et al. A reinforcement learning approach for MPPT control method of photovoltaic sources[J]. Renewable Energy, 2017, 108: 461-473.

[11] Duan J, Shi D, Diao R, et al. Deep-reinforcement-learning-based autonomous voltage control for power grid operations[J]. IEEE Transactions on Power Systems, 2019, 35(1): 814-817.

[12] Xu H, Domínguez-García A D, Sauer P W. Optimal tap setting of voltage regulation transformers using batch reinforcement learning[J]. IEEE Transactions on Power Systems, 2019, 35(3): 1990-2001.

[13] Le T T T, Moh S. An energy-efficient topology control algorithm based on reinforcement learning for wireless sensor networks[J]. Int. J. Control Autom, 2017, 10(5): 233-244.

[14] Le T T T, Moh S. Reinforcement-learning-based topology control for wireless sensor networks[J]. Proceedings of the Grid and Distributed Computing, 2016, 2016: 22-7.

[15] Zhang D, Han X, Deng C. Review on the research and practice of deep learning and reinforcement learning in smart grids[J]. CSEE Journal of Power and Energy Systems, 2018, 4(3): 362-370.

[16] Case D U. Analysis of the cyber attack on the Ukrainian power grid[J]. Electricity Information Sharing and Analysis Center (E-ISAC), 2016, 388(1-29): 3.

[17] Zhang H, Chen H, Xiao C, et al. Robust deep reinforcement learning against adversarial perturbations on state observations[J]. Advances in Neural Information Processing Systems, 2020, 33: 21024-21037.

[18] Ilahi I, Usama M, Qadir J, et al. Challenges and countermeasures for adversarial attacks on deep reinforcement learning[J]. IEEE Transactions on Artificial Intelligence, 2021, 3(2): 90-109.

[19] Kos J, Song D. Delving into adversarial attacks on deep policies[J]. arXiv preprint arXiv:1705.06452, 2017.

[20] Pinto L, Davidson J, Sukthankar R, et al. Robust adversarial reinforcement learning[C]//International Conference on Machine Learning. PMLR, 2017: 2817-2826.

[21] Wang J, Liu Y, Li B. Reinforcement learning with perturbed rewards[C]//Proceedings of the AAAI

conference on artificial intelligence. 2020, 34(04): 6202-6209.

[22] Smirnova E, Dohmatob E, Mary J. Distributionally robust reinforcement learning[J]. arXiv preprint arXiv:1902.08708, 2019.

[23] Zhou B, Zeng H, Liu Y, et al. Action set based policy optimization for safe power grid management[C]//Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part V 21. Springer International Publishing, 2021: 168-181.

[24] Zeng L, Sun M, Wan X, et al. Physics-constrained vulnerability assessment of deep reinforcement learning-based SCOPF[J]. IEEE Transactions on Power Systems, 2022.

[25] Xu P, Duan J, Zhang J, et al. Active power correction strategies based on deep reinforcement learning—Part I: A simulation-driven solution for robustness[J]. CSEE Journal of Power and Energy Systems, 2021, 8(4): 1122-1133.

[26] Pan A, Lee Y, Zhang H, et al. Improving robustness of reinforcement learning for power system control with adversarial training[J]. arXiv preprint arXiv:2110.08956, 2021.

[27] Zhang H, Chen H, Boning D, et al. Robust reinforcement learning on state observations with learned optimal adversary[J]. arXiv preprint arXiv:2101.08452, 2021.

[28] Espeholt L, Soyer H, Munos R, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures[C]//International conference on machine learning. PMLR, 2018: 1407-1416.

[29] Dai Z, Yang Z, Yang Y, et al. Transformer-xl: Attentive language models beyond a fixed-length context[J]. arXiv preprint arXiv:1901.02860, 2019.

[30] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[31] Parisotto E, Song F, Rae J, et al. Stabilizing transformers for reinforcement learning[C]//International conference on machine learning. PMLR, 2020: 7487-7498.

[32] Mishra N, Rohaninejad M, Chen X, et al. A simple neural attentive meta-learner[J]. arXiv preprint arXiv:1707.03141, 2017.