

# Probabilistic Charging Power Forecast of EVCS: Reinforcement Learning Assisted Deep Learning Approach

Yuanzheng Li <sup>ID</sup>, Member, IEEE, Shangyang He, Yang Li <sup>ID</sup>, Senior Member, IEEE, Leijiao Ge <sup>ID</sup>, Senior Member, IEEE, Suhua Lou <sup>ID</sup>, and Zhigang Zeng <sup>ID</sup>, Fellow, IEEE

**Abstract**—The electric vehicle (EV) and electric vehicle charging station (EVCS) have been widely deployed with the development of large-scale transportation electrifications. However, since charging behaviors of EVs show large uncertainties, the forecasting of EVCS charging power is non-trivial. This paper tackles this issue by proposing a reinforcement learning assisted deep learning framework for the probabilistic EVCS charging power forecasting to capture its uncertainties. Since the EVCS charging power data are not standard time-series data like electricity load, they are first converted to the time-series format. On this basis, one of the most popular deep learning models, the long short-term memory (LSTM) is used and trained to obtain the point forecast of EVCS charging power. To further capture the forecast uncertainty, a Markov decision process (MDP) is employed to model the change of LSTM cell states, which is solved by our proposed adaptive exploration proximal policy optimization (AePPO) algorithm based on reinforcement learning. Finally, experiments are carried out on the real EVCSs charging data from Caltech, and Jet Propulsion Laboratory, USA, respectively. The results and comparative analysis verify the effectiveness and outperformance of our proposed framework.

**Index Terms**—Deep learning, electric vehicle charging station, forecast uncertainty, probabilistic charging power forecasting, reinforcement learning.

## I. INTRODUCTION

A S ONE of the most essential actions, the carbon emission reduction has been adopted to address the problem of

Manuscript received 31 January 2022; revised 17 March 2022 and 5 April 2022; accepted 7 April 2022. Date of publication 19 April 2022; date of current version 23 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62073148, and in part by Tencent Rhinoceros Foundation of China under Grant CCF-Tencent RAGR20210102. (*Corresponding author: Yang Li*)

Yuanzheng Li and Zhigang Zeng are with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yuanzheng\_li@hust.edu.cn; zgeng@hust.edu.cn).

Shangyang He is with the China-EU Institute for Clean and Renewable Energy, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: heshangyang10@hust.edu.cn).

Yang Li is with the School of Electrical Engineering, Northeast Electric Power University, Jilin 132012, China (e-mail: liyang@neepu.edu.cn; liyangneda@gmail.com).

Leijiao Ge is with the Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin 300072, China (e-mail: legendlj99@tju.edu.cn).

Suhua Lou is with the School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: shlou@mail.hust.edu.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIV.2022.3168577>.

Digital Object Identifier 10.1109/TIV.2022.3168577

climate change. Therefore, renewable energy sources, such as wind and solar energy, are widely deployed to substitute fossil fuels. Besides, the electrification of transportation systems is another effective approach to reduce carbon emissions [1]. Due to the development of manufacture and related key technologies, the reliability and efficiency of electric vehicles (EVs) are significantly improved. Therefore, various countries aim to achieve a high EV penetration in the near future. It is predicted that the number of EVs will increase continually and may approximately reach 35 million at the end of 2022 throughout the world [2].

However, as one of the most important factors to expand the EV adoption, EV charging has brought great challenges to the secure and economic operations of the EV charging station (EVCS). Because the charging behavior of EV is with the nature of intrinsic randomness, the charging power of EVCS is uncertain [3]. Consequently, it is considered as a volatile and uncertain electricity load. Fluctuations of such load may threaten the operational security of EVCS and corresponding power systems [4]. To prevent these issues, it is crucial to well predict the EVCS charging power.

Current works about EVCS or EV charging power forecasting can be divided into two main categories, namely, the model-based and data-driven approaches. Different from the wind power and residential load power forecasting that are mainly influenced by the natural environment and the living habits of residents, the charging behavior of the EV users is the focus on the first category. Ref. [5] has employed the trip chain to establish a spatial-temporal behavior model of EVs, which is beneficial for forecasting their charging power. Considering the impacts of traffic on the charging behavior, Ref. [6] has provided a reliable approach for EVCS charging power forecasting. Based on the case in Shenzhen, China, the charging behaviors of EVs are considered for systematic forecasting of EV charging power [7].

Thanks to the advent of the cloud services and the Internet of Things, massive charging process data, such as the charging time and energy of EVs, can be collected. Thus, data-driven algorithms could be used for the forecast of EVCS charging power [8]. *Some classical data-driven forecasting algorithms are widely employed.* For instance, [9] has applied an autoregressive integrated moving average (ARIMA) model to forecast the charging power of massive EVCSs, which were distributed in Washington State and San Diego. Ref. [10] has combined

the least squares support vector machine algorithm and fuzzy clustering for the EVCS charging power forecasting task. Based on the historical data of Nebraska, USA, the effectiveness of the extreme gradient boosting on the charging power forecasting has been validated in [11].

The above algorithms rely on the structured input data with a specific human-defined feature, which requires cumbersome engineering-based efforts [12]. Hence, the deep learning method [13] becomes more popular in this field. For instance, Ref. [12] has reviewed numerous deep learning methods for EV charging load forecasting, and concluded that the long short-term memory (LSTM) model could reduce by about 30% forecasting error compared with the conventional artificial neural networks (ANN) model. *Considering the temporal dependency of the data, Ref. [14] has implemented the LSTM to capture the peak of charging power and achieve an effective result.* Ref. [15] has introduced the LSTM network to build a hybrid model for the forecasting of EVCS charging power and the experimental results demonstrated its effectiveness. Besides, compared with ARIMA and ANN, Ref. [16] has also illustrated the advantage of LSTM with abundant experiments.

Note that the above algorithms have been used to conduct the point forecasting, which only provides an expected charging power of EVCS in the future. Therefore, the probabilistic forecasting is introduced. It could forecast the probabilistic distribution of the future charging power, and thus provide more information, i.e., the expected value and the forecast uncertainties [17] [18]. Consequently, the probabilistic forecasting of EVCS charging power is conducted in this paper. To the best of our knowledge, there exist only a few works about this topic. For instance, Ref. [17] has applied four quantiles regression algorithms to the probabilistic EV load forecasting. In addition, the deep learning method is used in Ref. [18] to capture uncertainties of the probabilistic forecasting.

However, the above quantiles regression and deep learning-based methods have disadvantages. Regarding the first method, several steps are needed to be taken under different quantiles to obtain the forecast distribution, which would bring about much computational burden [17]. For deep learning, the forecast uncertainty is mainly caused by the model training and input data [18], [19]. Indeed, the stochastic parameters are normally introduced in the deep learning model, and the uncertainty is estimated through statistical indicators of its outputs. Therefore, the uncertainty of probabilistic forecast is usually obtained by using adequate samples, which needs the repeated running of the deep learning model and the usage of input data. In other words, traditional approaches may lead to a huge computation complexity.

Consequently, we propose an approach to solve the above issue. That is, a reinforcement learning assisted deep learning forecast approach is adopted to directly obtain the forecast uncertainty. This approach is designed to calculate the uncertainty once only, instead of repeated running. In detail, the LSTM is used to obtain the point prediction results of EVCS charging power, which is deemed as the expected value of the forecasting probabilistic distribution. Furthermore, note that the cell state of LSTM is one of the core inherent parameters, which could

represent the important information of the model and the input data, simultaneously. Therefore, the forecast uncertainty can be obtained from the cell state (this issue is explained in Section IV). As such state is varying from the input data, the forecast uncertainty also changes with the time-series data. Then, this change could be modelled as a Markov decision process (MDP). In this way, we innovatively introduce the reinforcement learning algorithm to assist the LSTM in order to obtain the forecast uncertainty. The reason for selecting reinforcement learning is that it is a powerful technique to improve the artificial target by the autonomous learning without using any prior knowledge [20]. Here, the target is to obtain the effective probabilistic forecast, and the reinforcement learning is used to learn the forecast uncertainty by observing cell states of the LSTM.

The contributions of this paper are shown as follows:

- 1) A reinforcement learning assisted deep learning framework is proposed for probabilistic EVCS charging power forecasting. To the authors' best knowledge, this is the first paper that uses the reinforcement learning algorithm to obtain forecast uncertainty in the field of EV charging power forecasting.
- 2) We model the variation of LSTM cell state as a MDP, which is solved by proximal policy optimization (PPO) to obtain the forecast uncertainty. In this case, the expected value of the EVCS charging power is forecasted by the LSTM, while the forecast uncertainty are yielded by the PPO.
- 3) An adaptive exploration PPO (AePPO) is further proposed. This would help adaptively balance the exploration and the exploitation during the training of PPO, which could improve its performance and help prevent it from getting trapped into local optima.
- 4) A data transformer method is developed to obtain the information of EVCS from distributed charger recordings, i.e., charging sessions for the LSTM training. It aggregates the charging sessions and transforms them into the time-series format that includes the information of charging process, i.e., the charging power, utilization time and demand satisfaction rate.

The remainder of this paper is organized as follows. Section II presents our proposed framework for EVCS charging power forecasting. In Section III, a data transformer method is proposed to preprocess the charging data. Section IV introduces the LSTM, the MDP model of the cell state variation and the original PPO. In Section V, the AePPO is proposed, and the case studies are conducted in Section VI. Finally, Section VII concludes this paper.

## II. FRAMEWORK

### A. Problem Formulation

The target of our studied probabilistic forecast problem is modelled by following formulations.

$$\tilde{F}_{t+1} = \hat{E}_{t+1} + \epsilon \quad (1)$$

$$\epsilon \sim \mathcal{N}(0, \delta_{t+1}) \quad (2)$$

$$\hat{E}_{t+1} = f(X_t) \quad (3)$$

$$\delta_{t+1} = g(f, X_t) \quad (4)$$

where  $\tilde{F}_{t+1}$  is the probabilistic forecast of EVCS charging power at time  $t + 1$ .  $\hat{E}_{t+1}$  is the expectation of the forecast values, which is obtained by the point forecast function  $f(\cdot)$ , as shown in (3) [19].  $\epsilon$  represents the noise, which stands for the uncertainty between the forecast and real values. In this paper, it is assumed that the  $\epsilon$  is Gaussian distributed with variance  $\delta_{t+1}$ , which is obtained by the variance forecast function  $g(\cdot)$ , presented in (4). It has been proved that the time series models based on the Gaussian distribution assumption can be applied to obtain a satisfactory performance [21]. Note that  $f(\cdot)$  and  $g(\cdot)$  represent the LSTM and our proposed AePPO, respectively. In addition,  $X_t = \{x_{t-T}, \dots, x_{t-1}, x_t\}$  is the vector of input features of the two functions. It stands for the features of  $T$  timestamps before  $t + 1$ , whereas  $x_t \sim \mathbf{N}^{N_f}$  ( $\mathbf{N}^{N_f}$  denotes the  $N_f$  dimensional integer space).

Generally,  $f(\cdot)$  provides a point forecast which represents the mean value of our target [22]. However, the obtained results might be unreliable since its inherent parameters are learned by the stochastic gradient descent optimization method such as Adam [18] and its input data  $X_t$  is related to the stochastic situation, i.e., the charging behavior of EV users. Therefore, it is necessary to quantify the uncertainty of  $f(\cdot)$  and  $X_t$ , and the introduction of the noise  $\epsilon$  is one of the most effective approaches [21], [23].

Usually, the  $\delta_{t+1}$  is forecasted through statistical indicators, which requires a massive repeated running of  $f(X_t)$  and thus leads to a huge computation complexity [19]. Consequently, it is expected to design a function  $g(\cdot)$  to forecast the variance  $\delta_{t+1}$  without the repeated running. The  $g(\cdot)$  should have the ability to capture the uncertainty of  $f(\cdot)$  and  $X_t$ , which means the input of  $g(\cdot)$  has to contain the characteristics of both point forecast model and data. Compared with other deep learning algorithms, such as ANN and conventional neural network, the LSTM owns a unique structural parameter, i.e., the cell state, which is the core parameter of LSTM and is also determined by the  $X_t$ . Therefore, the uncertainties coming from both  $f(\cdot)$  and  $X_t$  would make changes to the value of cell state, which could be captured by the  $g(\cdot)$  to forecast  $\delta_{t+1}$ . Besides, since the value of cell state is affected by its previous value, the change is modelled as a MDP in this paper. Therefore, the cell state is denoted as the state of the MDP, and the  $\delta_{t+1}$  is defined as the action. In this way, a reinforcement learning model, AePPO, is developed as  $g(\cdot)$  to observe the cell state and generate the action, i.e., the  $\delta_{t+1}$ . Generally, when the LSTM is applied to produce  $\hat{E}_{t+1}$ , the cell state would be produced, and the AePPO would capture it to determine  $\delta_{t+1}$ . Then, the probabilistic forecast can be constructed by  $\hat{E}_{t+1}$  and  $\delta_{t+1}$ .

## B. The Probabilistic Forecast Framework of EVCS Charging Power

As discussed above, the LSTM and AePPO are trained and utilized to obtain the mean and variance values for obtaining the predicted probabilistic distribution of EVCS charging power. In order to implement this procedure, we propose a probabilistic

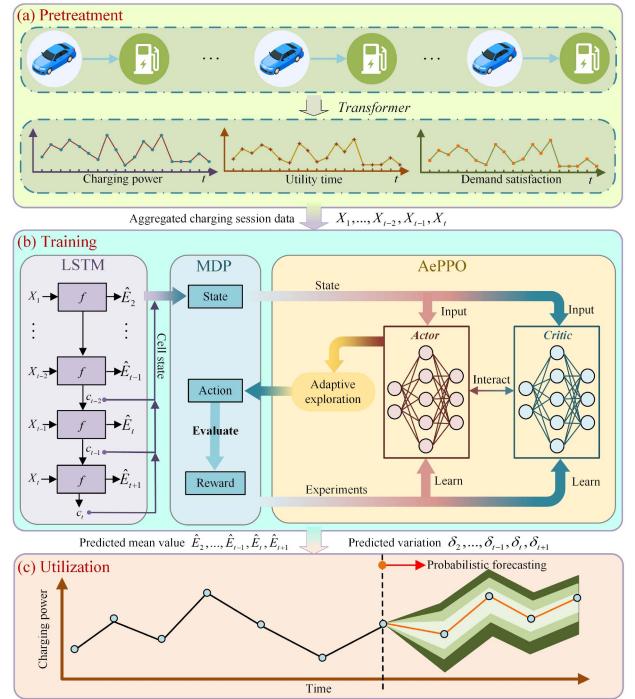


Fig. 1. The overall structure of the proposed forecast framework.

forecast framework, as shown in Fig. 1. It contains three parts, preprocessing, training, and utilization. They are presented by the three sub-figures in Fig. 1, which are denoted as (a), (b) and (c), respectively.

As illustrated in Fig. 1(a), an EVCS has numerous chargers which record the information about charging power in the period between the arrival and departure of an EV user [17]. Since all chargers may work simultaneously, the charging information of EVCS is obtained by recordings of chargers. In this paper, such recording is termed as the charging session. It consists of the period during the process of EV charging, i.e., the arrival and departure time of the EV, and the related data about the energy information, for instance, the demand and remaining energy of the corresponding EV. Note that one charging session only contains partial charging information of EVCS and the period between different charging sessions may overlap. The information in charging session during the overlap period should be gathered to derive the data of EVCS for LSTM training. In this case, the charging session data should be pre-processed by aggregating the information and transforming them into the time-series format with different features, such as the charging power, utility time and demand satisfaction rate.

During the training process shown in Fig. 1(b), the LSTM and AePPO are separately trained. At first, the EVCS data is used to train LSTM. Then, for the well-trained LSTM, its variation of cell state is modeled as a MDP. By doing so, AePPO could produce the action and interact with the LSTM according to the state. Afterwards, the reward is calculated based on the state and action, which is used for the training of AePPO.

Finally, the last part of this framework is the utilization, as shown in Fig. 1(c). The LSTM and AePPO models are utilized here. That is, the LSTM provides the mean value of probabilistic

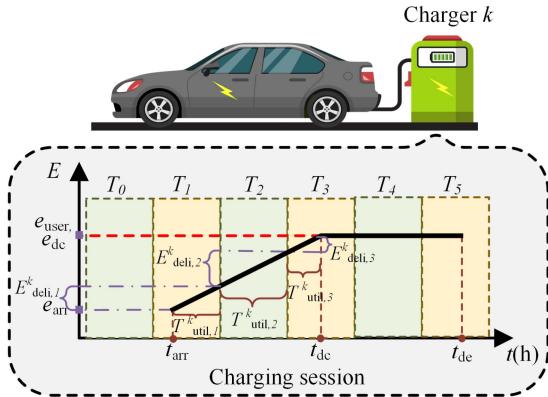


Fig. 2. The charging session recorded by charger  $k$ .

forecast distribution  $\hat{E}_{t+1}$ , while AePPO is used to determine the variation  $\delta_{t+1}$  based on the cell state of LSTM. In this way, the predicted probabilistic distribution  $\tilde{F}_{t+1}$  could be obtained by (1).

### III. DATA TRANSFORMER METHOD

In this section, the data transformer method is introduced to extract the charging information of EVCS from charging sessions, including the delivery energy, utility time and demand satisfaction rate. Then, they are aggregated into the time-series format for the training of LSTM.

Usually, the charging session data is composed by a series of time and energy information which is recorded during the charging process of the EV. An example of a charging session recorded by the  $k$ th charger is illustrated in Fig. 2. The time information includes  $t_{\text{arr}}$ ,  $t_{\text{dc}}$  and  $t_{\text{de}}$ , which denote the time when EV is connected to the charger, done charging and departure. Besides, the energy information is also recorded, including  $e_{\text{arr}}$ ,  $e_{\text{dc}}$  and  $e_{\text{de}}$ , which represent the remaining energy in the EV battery at  $t_{\text{arr}}$ ,  $t_{\text{dc}}$  and  $t_{\text{de}}$ , respectively. Furthermore, the charging session also contains the user demand, termed as  $e_{\text{user}}$ . Note that expect the above information, the energy  $e_t$  at the end of timestamp  $t$  is also recorded.

However, it is difficult to use the original charging session data for the forecasting of EVCS charging power. The charging session focuses on a single EV, which only contains partial information of the EVCS since numerous EVs may charge at the same time.

Therefore, the charging session is manually split by timestamps, and each of the timestamps represents 1 h, as illustrated in Fig. 2. Three features are defined to summarize the charging information in each timestamp  $t$ , i.e., the delivery energy  $E_{\text{deli},t}^k$ , the demand satisfaction rate  $D_t^k$  and the utilization time  $T_{\text{util},t}^k$ . They are formulated as follows.

$$E_{\text{deli},t}^k = \begin{cases} e_t - e_{\text{arr}}, & t_{\text{arr}} < t < t_{\text{arr}} + 1 \\ e_t - e_{t-1}, & t_{\text{arr}} + 1 \leq t < t_{\text{dc}} \\ 0, & t_{\text{dc}} \leq t < t_{\text{de}} \end{cases} \quad (5)$$

$$T_{\text{util},t}^k = \begin{cases} t - t_{\text{arr}}, & t_{\text{arr}} < t < t_{\text{arr}} + 1 \\ 1, & t_{\text{arr}} + 1 \leq t < t_{\text{de}} \end{cases} \quad (6)$$

$$D_t^k = \frac{E_{\text{deli},t}^k}{e_{\text{user}} - e_{\text{arr}}} \times 100\% \quad (7)$$

where  $k$  indicates the charger. Examples of these features are shown in Fig. 2, where the charging session is split into 6 timestamps.

Then, considering  $N_c$  chargers are set in the EVCS, the features split by timestamp are aggregated by the following equations:

$$E_t = \sum_{k=1}^{N_c} E_{\text{deli},t}^k, \quad T_t = \sum_{k=1}^{N_c} T_{\text{util},t}^k, \quad D_t = \sum_{k=1}^{N_c} D_t^k \quad (8)$$

After aggregation, the three features are used for the training of LSTM. They could summarize the information of time and energy contained in the charging session data. On the one hand, the delivery energy measures the service quality of the EVCS in terms of time. On the other hand, the time information of the charging process is summarized by the demand satisfaction rate and utilization time. In this way, they are selected as the features for the training of LSTM, aiming to promote its performance. Therefore, the number of features  $N_f$  is 3 in this paper, and the input feature vector at timestamp  $t$  is denoted by  $x_t = [E_t, T_t, D_t]$ .

### IV. REINFORCEMENT LEARNING ASSISTED DEEP LEARNING ALGORITHM

This section presents the reinforcement learning assisted deep learning algorithm. In the first subsection, the LSTM is introduced to forecast the mean value of charging power with the aggregated time-series data. Then, the variation of the LSTM cell state is modeled as a MDP in the second subsection. In the last subsection, the PPO is used to solve the MDP, and obtain the variance of the forecasts.

#### A. Long Short Term Memory

In this paper, in order to obtain the mean value of EVCS charging power for the forecast of probabilistic distribution, the LSTM is applied. Compared with other deep learning algorithms, such as the fully connected network, convolutional neural network and vanilla recurrent neural network, LSTM is more capable of learning the long-term dependencies inherent to the time-series data and would not suffer from the vanishing gradients [23]. Besides, Overall, the LSTM is formulated as follows.

$$y_{t+1} = A(X_t, y_t, c_t) \quad (9)$$

where  $y_{t+1}$  represents the output of LSTM at time  $t + 1$ , which stands for the predicted mean value of EVCS charging power. The LSTM model A takes  $X_t$ ,  $y_t$  and  $c_t$  as inputs, which denote the input data, the output and the cell state of LSTM at time  $t$ , respectively.

For clarity, the formulations of the LSTM is referred as follows.

$$i_{t+1} = \sigma(W_i \cdot [y_t, X_t] + b_i) \quad (10)$$

$$f_{t+1} = \sigma(W_f \cdot [y_t, X_t] + b_f) \quad (11)$$

$$o_{t+1} = \sigma(W_o \cdot [y_t, X_t] + b_o) \quad (12)$$

$$\tilde{c}_{t+1} = \tanh(W_c \cdot [y_t + X_t] + b_c) \quad (13)$$

$$c_{t+1} = f_{t+1} \odot c_t + i_{t+1} \odot \tilde{c}_{t+1} \quad (14)$$

$$y_{t+1} = o_{t+1} \odot \tanh(c_{t+1}) \quad (15)$$

where  $[W_i, W_f, W_o, W_c]$  are the learnable weight metrics and  $[b_i, b_f, b_o, b_c]$  are the learnable bias metric.  $\sigma$ ,  $\tanh$  are the sigmoid and hyperbolic tangent activation functions. Their output range is  $(0,1)$ .  $\odot$  indicates the elementwise multiplication.  $i_{t+1}, f_{t+1}, o_{t+1}$  stand for the input, forget and output gate, respectively.  $\tilde{c}_{t+1}$  represents the candidate information of the inputs in  $c_{t+1}$ , i.e., the cell state.

It can be seen from (14) and (15) that the value of the cell state highly influences  $y_{t+1}$ . Besides, these equations also show that the cell state is determined by the learnable parameters and the input data  $X_t$ , and they are affected by the uncertainty from the model and user's behaviour, respectively. Wherein, the uncertainty of the LSTM model comes from the learnable parameters, which are determined by the stochastic gradient descent optimization method. In addition, since  $X_t$  is aggregated by the charging session data, it is uncertain as well because of correlations with the behaviour of users.

### B. The Modeling of LSTM Cell State Variation

As described above, the input of LSTM  $X_t$  is the time-series format, which reflects the behavior of the users. Besides, because of the recurrent structure of the LSTM, the cell state is influenced by its previous value. Therefore, the variation of the LSTM cell state is caused by the uncertainty that comes from both the model and the data. In this case, the variation of the cell state should be modelled to extract the variance of the probabilistic forecast distribution from the LSTM cell state. Since the cell state is determined by its previous state, its variation can be modelled as a MDP. Normally, the MDP is represented by a tuple  $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ .  $\mathcal{S}$  is the state space that stands for all possible states of the environment.  $\mathcal{A}$  represents the action space, i.e., the agent interacts with the environment to produce action  $a \in \mathcal{A}$  for the guidance of state transition.  $\mathcal{P} = \{p(s_{t+1}|s_t, a_t)\}$  stands for the set of transition probability, and  $\mathcal{R} = r(s_t, a_t)$ ,  $\mathcal{R} \in \mathbb{R}$ ;  $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  depends on the state and action and is termed as the reward. In our problem, the MDP can be formulated as follows:

- 1) *Environment*: The environment produces the state  $s_t$  and computes the reward  $r_t$  based on the action  $a_t$  of the agent. In this paper, the variance of the probabilistic forecast distribution is obtained from the variation of the cell state, which is the most representative parameter of the LSTM. *The definition requires a premise, namely, the LSTM model is well-trained and its learnable parameters are fixed. The promise ensures the decisive position of the cell state in the model, therefore, a well-trained LSTM is defined as the environment of MDP.*
- 2) *Agent*: Here, the agent stands for a policy to obtain the action from observing the state. Besides, the agent could imply the self-learning from its experience to gradually improve the policy.

3) *State*: The definition of the state  $s_t$  at time  $t$  should follow two criteria. First, the state should be the most representative parameter of the environment. Second, the state should only be related to the previous state and not affected by the next state, namely, the non-aftereffect property. According to the calculation process of LSTM shown in (10)–(15), the most suitable parameter is the cell state, which not only determines the output of LSTM but also be obtained by the cell state of the previous time. Therefore,  $s_t$  is set as  $c_t$  and the state space equals to all the possible value of  $c_t$ .

4) *Action*: As the uncertainty of the model and input data are contained in  $s_t$ , it is expected that the agent can capture this uncertainty to produce the variance of the forecast distribution, i.e., the action  $a_t = \delta_t$ .

5) *Reward*: The reward  $r(s_t, a_t)$  indicates the evaluation index of  $a_t$  on the basis of the  $s_t$ . It gives feedback to the agent about the performance of its action, which could be used to update the agent. Since our target is to obtain the probabilistic distribution of future charging power, which is constructed by the output of the environment, i.e., LSTM and the agent's action. In our case, the reward is designed to evaluate the performance of the predicted distribution.

To quantify it, the continuous ranked probability score (CRPS) is introduced to measure of performance for probabilistic forecasts, i.e., the accuracy of the predicted probabilistic distribution [24]. The details of CRPS calculation are given in the Appendix A.

Note that the lower CRPS represents the more accurate probabilistic forecast. Therefore, the reward is defined as  $r = -\zeta \times \text{CRPS}$ , which should be maximized by the agent.  $\zeta$  stands for the shrinkage coefficient, which is set as 0.75.

### C. Proximal Policy Optimization

Based on the MDP, a recent reinforcement learning algorithm, called PPO, is introduced to train the agent for generating the optimal policy  $\pi(a_k|s_k)$ . The PPO contains two types of deep neural networks, which are termed as *actor* and *critic*. The *actor* works as the agent, and it produces the action by a policy  $\pi$  with parameter  $\theta^\pi$ . On the other hand, the *critic* is used to evaluate the performance of *actor*, parametrized by  $\theta^Q$ . The *critic* approximates the value function  $V^{\pi_k}(s_k)$ , which evaluates the value of the state  $s_k$ , i.e., it represents the expectation of the reward from  $k$  to the end of the MDP, which can be formulated as follows.

$$V^{\pi_k}(s) = E_\pi [U_k | S_k = s] \quad (16)$$

where  $U_k$  is the return function, which is defined as the discounted cumulative reward from state  $s_k$ .

$$U_k(s_k) = \sum_{t=k}^T \gamma \times r_t \quad (17)$$

where  $\gamma \in [0, 1]$  stands for the discount factor,  $r_t$  denotes the value of reward and  $T$  is the finite time horizontal of the MDP.

The main target of the PPO is to train the *actor* and *critic*, by learning from the experience tuples  $\langle s_k, a_k, r_k, s_{k+1} \rangle$ . It is obtained by the interaction between the *actor* and the environment. The *actor* generates a Normal distribution  $\mathcal{N}(a_{k,\text{mean}}, a_{k,\text{var}})$  after observing  $s_k$ . That is, the action  $a_k$  is sampled from this distribution, i.e.,  $a_k \sim \mathcal{N}(a_{k,\text{mean}}, a_{k,\text{var}})$ . Therefore, this introduces randomness in the produced action and thus creates more diverse actions when observing the same state. In this way, it leads to the diversity of rewards since they are related to the action. This mechanism is termed as the exploration of PPO, which is used to enrich the experience tuple in order to prevent the actor from getting trapped in local optima.

Then, based on the prediction of the mean value  $y_k$  obtained by LSTM, the predicted probabilistic distribution  $\mathcal{N}(y_k, a_k)$  is constructed. Afterwards,  $r_k$  could be calculated by the reward function to evaluate  $\mathcal{N}(y_k, a_k)$ , and the state at the next time  $s_{k+1}$  is also restored. With the experience tuple, the parameters of both networks are updated by following equations.

$$\theta_{k+1}^\pi = \theta_k^\pi + \eta_\pi \nabla_{s,a \sim \pi_k} \mathcal{L}^{\text{CLIP}} \quad (18)$$

$$\theta_{k+1}^Q = \theta_k^Q + \eta_Q \nabla_{\theta Q_k} \mathcal{L}^Q \quad (19)$$

where  $\eta_\pi$  and  $\eta_Q$  denote the learning rates of *actor* and *critic*, respectively.  $\mathcal{L}^{\text{CLIP}}$  and  $\mathcal{L}^Q$  stand for the loss functions of the two networks, which are given by

$$\begin{aligned} \mathcal{L}^{\text{CLIP}} &= \mathbb{E}_{s,a \sim \pi_k} \left[ \min \left( \frac{\pi_k(a|s)}{\pi_{k-1}(a|s)} A_{s,a}^{\pi_k}, \right. \right. \\ &\quad \left. \left. \text{clip} \left( \frac{\pi_k(a|s)}{\pi_{k-1}(a|s)}, 1-\epsilon, 1+\epsilon \right) A_{s,a}^{\pi_k} \right) \right] \end{aligned} \quad (20)$$

$$\mathcal{L}^Q = \mathbb{E}_{s,a \sim \pi_k} [(\gamma V^{\pi_k}(s_{k+1}) + r(s_k, a_k) - V^{\pi_k}(s_k))^2] \quad (21)$$

where  $\pi_{k-1}(a|s)$  denotes the policy of  $(k-1)$ th iteration, and  $\gamma$  is the discount rate.  $\text{clip}(t, t_{\min}, t_{\max})$  is the clip function. It returns  $t_{\max}$  if  $t > t_{\max}$ , and  $t_{\min}$  if  $t < t_{\min}$ .  $A_{s,a}^{\pi_k}$  is the advantage of action  $a$  under state  $s$ , which is represented by the difference between  $a$  and the averaged performance of the *actor*, formulated as follows.

$$A_{s,a}^{\pi_k} = Q(s_k, a_k) - V^{\pi_k}(s_k) \quad (22)$$

The Q-function  $Q(s_k, a_k)$  evaluates the value of executing action  $a_k$  at state  $s_k$ , i.e., it represents the expectation of the reward from  $s_k$  to  $s_T$  after selecting the action  $a_k$ . The formulation of the  $Q$  is shown below.

$$Q(s, a) = E_\pi [U_k | S_k = s, A_k = a] \quad (23)$$

Then, the  $Q(s_k, a_k)$  is calculated by the following equation in the PPO algorithm.

$$Q(s_k, a_k) = r(s_k, a_k) + \sum_{i=1}^T \gamma^{T-i} r(s_{k-i}, a_{k-i}) \quad (24)$$

## V. ADAPTIVE EXPLORATION PROXIMAL POLICY OPTIMIZATION

The *actor* of PPO will generate  $a_{k,\text{mean}}$  and  $a_{k,\text{var}}$ , based on the state  $s_k$ . As discussed in the above section, the action  $a_k$  is sampled from the distribution  $\mathcal{N}(a_{k,\text{mean}}, a_{k,\text{var}})$ . The value of  $a_{k,\text{var}}$  determines the exploration of PPO. With the same  $a_{k,\text{mean}}$ , a larger  $a_{k,\text{var}}$  represents a wider distribution, which brings higher degree of exploration and more diverse experiences. On the contrary, the smaller  $a_{k,\text{var}}$  cannot provide a richer experience because the action distribution is more concentrated, and the probability of  $a_k$  deviating from  $a_{k,\text{mean}}$  is lower. Under this situation, the PPO trains its *actor* and *critic* according to its previously generated experience, which is termed as the process of exploitation.

Note that exploration and exploitation are usually contradictory. Ideally, the reinforcement learning algorithm focuses on its exploration in the early stage of the training to obtain sufficient experiences. However, if the algorithm keeps a high exploration, it is hard to be convergent because of its high randomness. In this case, the exploration will shrink according to the training. Thus, the degree of exploitation is increased. Consequently, the explored experiences should be fully utilized to help the *actor* and *critic* converge. However, the practical training process of PPO is different from the ideal one. Since  $a_{k,\text{var}}$  of PPO is generated by the *actor*, which means the degree of exploration is uncontrollable.  $a_{k,\text{var}}$  is dependent on both the initial condition (set manually) and the inherent parameters of *actor*, which is difficult to converge. Thus it may cause higher exploitation prematurely. The consequence of lacking enough exploration is the lower diversity of the experiences. In this way, the PPO may get trapped in the local optima.

To tackle this issue, we propose an adaptive exploration mechanism for PPO to dynamically balance its exploration and exploitation. This mechanism is termed as the adaptive exploration proximal policy optimization (AePPO). In the AePPO, the action is sampled from  $\mathcal{N}(a_{k,\text{mean}}, a_{k,\text{var}}^{\text{AePPO}})$  where  $a_{k,\text{mean}}$  is generated by the *actor* and  $a_{k,\text{var}}^{\text{AePPO}}$  is determined by our proposed adaptive exploration mechanism.

Using such a mechanism, the exploration of AePPO is increased in the earlier training stage and then is gradually shrunk to enhance the exploitation of the experiences. It is inadvisable to achieve this target by gradually reducing the value of  $a_{k,\text{var}}^{\text{AePPO}}$  according to the training. This is because different actions may produce the same performance, i.e., reward, which may limit the expansion of experiences. In this case, the  $a_{k,\text{var}}^{\text{AePPO}}$  should be adjusted according to the performance of the reward.

In PPO, the actions are sampled from  $\mathcal{N}(a_{k,\text{mean}}, a_{k,\text{var}}^{\text{AePPO}})$  and the reward is determined by both the action and state, i.e.,  $r_k = r(s_k, a_k)$ . Then, considering the Bayes' theorem, since  $r_k$  is one-to-one corresponds to  $(s_k, a_k)$  pair, when the  $s_k$  is fixed,  $r_k$  should subject to a Normal distribution  $\mathcal{N}(r_k^{\text{mean}}, r_k^{\text{var}})$  as well, and  $r_k^{\text{mean}}$  and  $r_k^{\text{var}}$  represent the mean and variance of the distribution, respectively. As shown in Fig. 3(a), the two parameters could be estimated by the following equations.

$$r_k^{\text{mean}} = \frac{1}{N_a} \sum_{j=1}^{N_a} r(s_k, a_k^j) \quad (25)$$

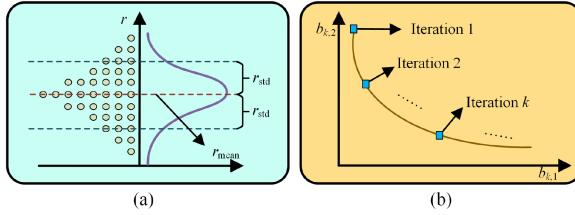


Fig. 3. (a) The sampling of reward; (b) The changes of parameter with respect to iterations.

$$r_k^{\text{var}} = \frac{1}{N_a} \sum_{j=1}^{N_a} (r(s_k, a_k^j) - r_k^{\text{mean}}) \quad (26)$$

$$a_k^j \in (a_k^1, a_k^2, \dots, a_k^i, \dots, a_k^{N_a}) \quad (27)$$

where  $(a_k^1, a_k^2, \dots, a_k^i, \dots, a_k^{N_a})$  indicates the  $N_a$  actions which are sampled from  $\mathcal{N}(a_{k,\text{mean}}, a_{k,\text{var}}^{\text{AePPO}})$ .

Note that  $r_k^{\text{var}}$  represents the exploration of the PPO and it is influenced by  $a_{k,\text{var}}^{\text{AePPO}}$ . If  $r_k^{\text{var}}$  is enlarged by changing  $a_{k,\text{var}}^{\text{AePPO}}$ , it indicates the diversity of experiences is enriched, i.e., the exploration ability of PPO is enhanced. Besides, since  $r_k^{\text{mean}}$  evaluates the performance of PPO, the higher  $r_k^{\text{mean}}$  means improved convergence as well. In this case, the focus on exploration and exploitation should be changed during the training. This process could be represented by the following equation.

$$f_k^{\text{exp}} = b_{k,1} r_k^{\text{mean}} + b_{k,2} r_k^{\text{var}} \quad (28)$$

where  $f_k^{\text{exp}}$  represents the degree of exploration.  $b_{k,1} < 1$  and  $b_{k,2} = 1 - \sqrt{1 - (b_{k,1} - 1)^2}$  denote two parameters which represent the focus on  $r_k^{\text{mean}}$  and  $r_k^{\text{var}}$ , respectively. As shown in Fig. 3(b), the values of these parameters vary according to the training iteration  $k$ . In the early stage of the training,  $b_{k,2}$  is larger than  $b_{k,1}$ , which means  $f_k^{\text{exp}}$  is highly influenced by  $r_k^{\text{var}}$ . Then,  $b_{k,1}$  is increased according to the training, and thus,  $r_k^{\text{var}}$  gradually decreases. When  $b_{k,1} > b_{k,2}$ , the value of  $f_k^{\text{exp}}$  is more influenced by  $r_k^{\text{var}}$ , which means the larger  $f_k^{\text{exp}}$  causes higher  $r_k^{\text{var}}$  and thus strengthen the exploration. On the other hand, when  $b_{k,1} \leq b_{k,2}$ , the raise of  $f_k^{\text{exp}}$  leads to the increase of  $r_k^{\text{mean}}$ , the focus of  $f_k^{\text{exp}}$  has changed from raising the exploration to increase the average performance of PPO, i.e., enhancing the exploitation. In this case,  $f_k^{\text{exp}}$  should be maximized by adjusting the  $a_{k,\text{var}}^{\text{AePPO}}$  during the training of AePPO, i.e.,  $-f_k^{\text{exp}}$  should be minimized. In this paper, it is solved by the particle swarm optimization (PSO), a well-known optimization algorithm [25].

In conclusion, after the separate training, the interconnection between LSTM and proposed AePPO is described as follows. As illustrated in the upper side of Fig. 4, LSTM takes the input data, which is obtained by the data transformer and performs its internal calculation according to the (10)–(15). After that, the output of LSTM is the predicted mean value  $\hat{E}_{k+1}$  of EVCS charging power. Then, the cell state of LSTM is extracted to fulfill the experience of AePPO, which serves as the state  $s_k$  of AePPO. The lower side of Fig. 4 shows the process of AePPO training iteration. The square with a yellow background in this figure represents the update mechanism of actor and

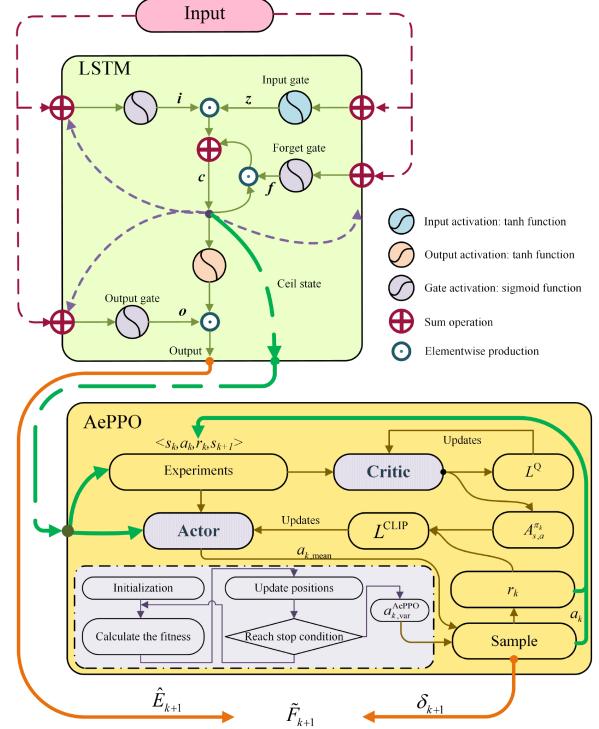


Fig. 4. The interconnection between LSTM and AePPO.

*critic* according to the (19)–(24). Besides, the area with a violet background represents the procedure of adaptive exploration. Then, the PSO is applied to minimize  $-f_k^{\text{exp}}$  and obtains  $a_{k,\text{var}}^{\text{AePPO}}$ , which represents the adaptive exploration of AePPO. Afterwards, the variation of predicted distribution  $\delta_{k+1}$  is sampled from  $\mathcal{N}(a_{k,\text{mean}}, a_{k,\text{var}}^{\text{AePPO}})$  where  $a_{k,\text{mean}}$  is determined by the *actor* of AePPO. Finally, based on (1)–(4), the probabilistic prediction distribution of EVCS charging power is represented by  $\tilde{F}_{k+1} = \mathcal{N}(\hat{E}_{k+1}, \delta_{k+1})$ . The pseudocode of LSTM-AePPO is provided in the Appendix B.

## VI. CASE STUDY

### A. Data Description and Experiential Initialization

In this part, we conduct a case study to verify the effectiveness of our proposed algorithm, i.e., LSTM-AePPO. The training data of the algorithm is collected from ACN dataset [26], which is an open dataset for EVCS charging researches. The charging sessions in ACN dataset are recorded from two EVCSs, one at Caltech, Pasadena, including 54 chargers and the other located in Jet Propulsion Laboratory (JPL), La Canada, USA, containing 50 chargers. In our experiments, the charging session from 1st June 2018 to 1st June 2020 are applied to train the LSTM-AePPO, in which 25916 and 22128 charging sessions are included in the case of Caltech and JPL, respectively.

In order to demonstrate the effectiveness of LSTM-AePPO, numerous algorithms, i.e., support vector quantile regression (SVQR) [27], linear quantile regression (QR) [28], and gradient boosting quantile regression (GBQR) [29] are introduced for comparisons. Considering the seasonal variations of charging

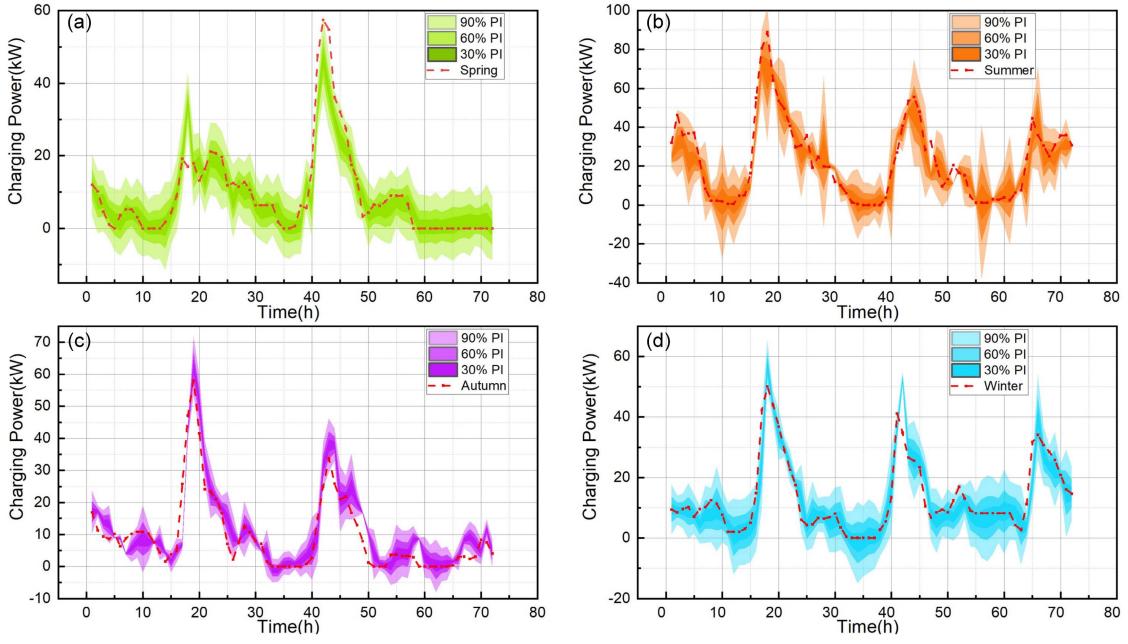


Fig. 5. The probabilistic forecasting of Caltech, obtained by the proposed LSTM-AePPO framework.

TABLE I  
THE HYPERPARAMETERS OF THE PROPOSED LSTM-AEPP0

Symbol	Description	Value
$\eta_a, \eta_Q$	The learning rate of actor and critic	0.0001
$\eta_{lstm}$	The learning rate of LSTM	0.001
$\epsilon$	The clip value of actor loss	0.1
$\gamma$	The discount factor	0.99
$N_e$	The maximum training number of AePPO	10000
$N_{ps}$	The population of PSO	20
$N_{var}$	The maximum iteration of PSO	100
$e_1, e_2$	The learning factor of PSO	2
$\omega$	The inertia weight of PSO	0.7

behavior, the performance and metric comparisons are conducted on the four seasons, respectively. Moreover, for further verifying the outperformance of our proposed AePPO, the traditional PPO is introduced to generate the  $\delta_{k+1}$  from the cell state for comparisons, which is termed as LSTM-PPO.

Then, CRPS, Winkler [23], and Pinball [30] are introduced as evaluation metrics. The three metrics focus on the accuracy, variation, and reliability of the forecast distribution, respectively. Their details are presented in Appendix C.

In this paper, the features of former  $N_h$  hours  $[x_{i-1}, \dots, x_{i-N_h}]$  are used in the probabilistic forecasting of EVCS charging power at hour  $i$ . The value of  $N_h$  is set as 12, and other hyperparameters of LSTM-AePPO are given in Table I. Note that the experiments are conducted on a computer with 8 GB RAM, Intel i5-8265 U CPU, and implemented in Python.

#### B. The Performance of Probabilistic Forecasting Obtained by LSTM-AePPO

The probabilistic forecast of EVCS charging power obtained by LSTM-AePPO are shown in Fig. 5 and Fig. 6, which illustrate the results on Caltech and JPL cases among 3 test days (72 hours). The subfigures (a), (b), (c) and (d) represent the forecast

of Spring, Summer, Autumn, and Winter, respectively. Specifically,  $p\%$  prediction interval (PI) is used to denote the interval between the lower and upper quantiles at probability  $p$ , and the darker color denotes the PI with a smaller  $p$ . Consequently, the 30% PI is shown with the darkest color while the 90% PI corresponds to the lightest. Note that the real charging power is represented by the red line in these figures.

One of the biggest challenges of the charging power forecasting is to predict the peak, which varies according to the seasons. For instance, in the Spring and Autumn case of Caltech, it could be learned that the peaks emerge in around the 18th and 42nd hours, as shown in Figs. 5(a) and (c). However, as shown in Figs. 5(b) and (d), the peaks during the Summer and Winter appear 3 times, which is more frequent. Besides, as shown in Fig. 6, the peaks are more irregular in the JPL case, which appear 2 times in the Spring and Autumn cases, 1 time in the Summer case and 3 times in the Winter case. Although the peak appears irregularly, LSTM-AePPO achieves a good result. The fluctuation of the probabilistic forecast distribution can capture the EVCS charging power. As shown in Figs. 5 and 6, most of the values in the red lines are covered by the 90% PI, which means that the probabilistic forecast distribution could represent the variation of real charging power.

#### C. Metrics Comparison Among Different Algorithms

To verify the effectiveness of LSTM-AePPO, the Winkler, Pinball and CRPS are introduced as metrics for comparison. Since the values of Winkler and Pinball are related to PI, the effectiveness of our proposed algorithm is demonstrated by comparing with other algorithms under 30%, 60% and 90% PIs, as shown in Table II.

In this table, the best metric value of each case is marked in bold. It could be seen that the LSTM-AePPO outperforms other

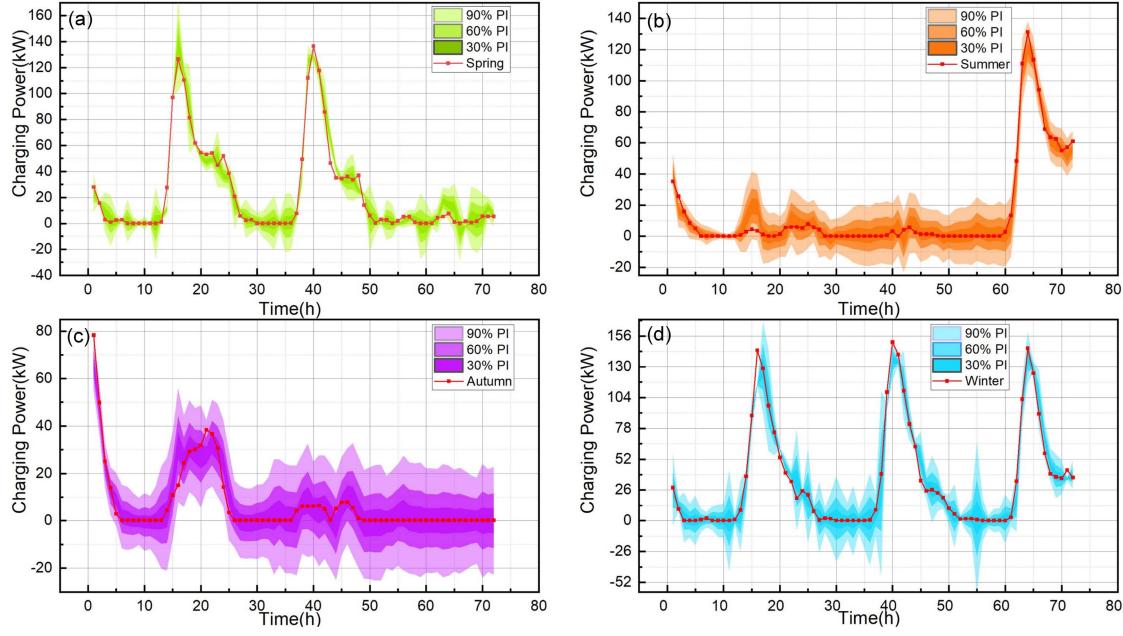


Fig. 6. The probabilistic forecasting of JPL obtained by the proposed LSTM-AePPO framework.

TABLE II  
THE SEASONAL METRIC COMPARISON OF TWO SITES UNDER DIFFERENT PI

		Winkler					Pinball					
		QR	QRSVM	GBQR	LSTM-PPO	LSTM-AePPO	QR	QRSVM	GBQR	LSTM-PPO	LSTM-AePPO	
Caltech	Spring	30	144.565	113.104	116.499	54.882	<b>30.099</b>	18.922	17.335	17.715	15.412	<b>15.158</b>
		60	85.686	75.070	73.747	45.429	<b>21.036</b>	18.023	16.762	17.095	18.512	<b>13.311</b>
		90	40.323	39.088	38.607	34.876	<b>23.966</b>	16.656	15.783	15.940	23.966	<b>14.057</b>
	Summer	30	203.099	160.325	191.611	65.351	<b>26.914</b>	26.366	<b>21.962</b>	24.434	24.424	23.475
		60	99.963	93.854	102.781	40.923	<b>16.665</b>	24.387	20.960	22.590	23.475	<b>15.835</b>
		90	44.193	45.280	45.503	27.592	<b>13.074</b>	21.359	18.935	20.003	21.815	<b>18.664</b>
	Autumn	30	131.399	83.484	93.647	37.247	<b>67.858</b>	18.495	16.616	18.910	22.154	<b>14.480</b>
		60	76.467	42.663	47.674	22.865	<b>35.791</b>	17.788	16.595	18.434	15.039	<b>14.683</b>
		90	34.970	28.657	28.326	<b>18.342</b>	21.510	16.446	15.818	16.988	19.850	<b>14.870</b>
	Winter	30	131.210	137.102	124.088	55.526	<b>19.342</b>	19.777	18.939	18.630	13.498	<b>13.328</b>
		60	76.665	87.910	77.862	45.158	<b>15.463</b>	18.984	18.101	17.884	17.004	<b>14.124</b>
		90	28.660	30.142	30.627	21.798	<b>14.666</b>	17.271	16.590	16.401	24.813	<b>14.378</b>
JPL	Spring	30	367.828	292.015	278.605	71.544	<b>27.800</b>	36.793	30.789	32.484	23.128	<b>18.072</b>
		60	171.225	157.093	135.559	26.264	<b>19.512</b>	34.360	29.845	30.738	31.896	<b>20.312</b>
		90	54.291	49.069	49.749	21.784	<b>18.296</b>	30.001	26.917	27.372	46.840	<b>20.760</b>
	Summer	30	210.133	237.387	208.853	37.048	<b>24.952</b>	32.200	26.701	27.833	25.464	<b>25.464</b>
		60	103.983	112.102	100.350	27.096	<b>21.368</b>	31.269	27.156	27.538	31.480	<b>22.680</b>
		90	52.745	49.958	50.535	24.824	<b>19.608</b>	27.807	25.061	25.105	37.528	<b>22.584</b>
	Autumn	30	157.871	123.457	94.360	29.240	<b>23.064</b>	21.857	21.287	21.325	22.712	<b>21.016</b>
		60	54.617	57.052	52.259	21.976	<b>19.416</b>	23.676	23.167	23.110	25.400	<b>21.464</b>
		90	47.000	47.033	47.040	18.232	<b>18.584</b>	22.834	22.420	22.123	28.856	<b>21.656</b>
	Winter	30	524.566	569.323	505.956	38.328	<b>32.440</b>	45.037	43.167	40.867	22.680	<b>22.136</b>
		60	258.636	314.319	265.607	23.416	<b>23.256</b>	41.205	39.586	37.381	24.920	<b>24.888</b>
		90	51.263	76.256	73.966	19.032	<b>18.776</b>	34.898	33.815	32.077	31.512	<b>29.400</b>

algorithms on Winkler and Pinball in most cases. For example, in the Spring case of Caltech, the Winkler and Pinball values of LSTM-AePPO at 30%, 60% and 90% PIs are (30.099, 15.158), (21.036, 13.311) and (23.966, 14.057), which are lower than the Winkler values of other algorithms. Besides, in the Summer and Autumn cases of Caltech, the LSTM-AePPO always performs the best at 60% PI. Moreover, when focusing on the Winter case of Caltech, the QR, QRSVM, GBQR and LSTM-PPO also perform worse than LSTM-AePPO since their Winkler and Pinball values are larger. Generally, the forecast distribution obtained by LSTM-AePPO have a better variation and reliability at Caltech case. In addition, the effectiveness is also verified

comprehensively in the JPL case. As shown in the Table II, the Winkler and Pinball numbers obtained by LSTM-AePPO in the JPL case are bold, which means QR, QRSVM, GBQR and LSTM-PPO are all surpassed from both the variation and reliability of the forecast distribution.

Note that the LSTM-AePPO does not perform the best in the following two cases, i.e., the Winkler of Caltech Autumn case at 90% PI (21.510 for LSTM-AePPO but 18.342 for LSTM-PPO) and the Pinball of Caltech Summer case at 30% PI (23.475 for LSTM-AePPO but 21.962 for QRSVM). The cause of this phenomenon may be the correlation between Winkler and Pinball to the PI. Therefore, as a metric that is independent of PI, the CRPS

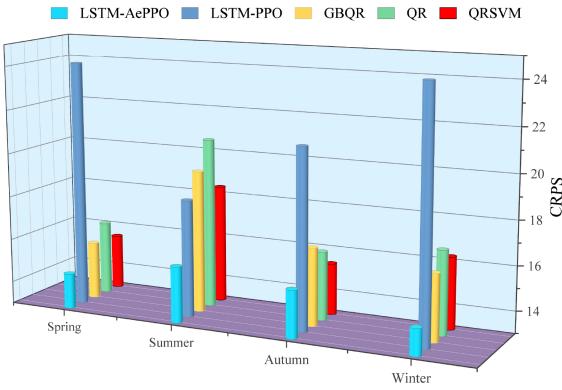


Fig. 7. The comparisons of CRPS in the case of Caltech among the five mentioned algorithms.

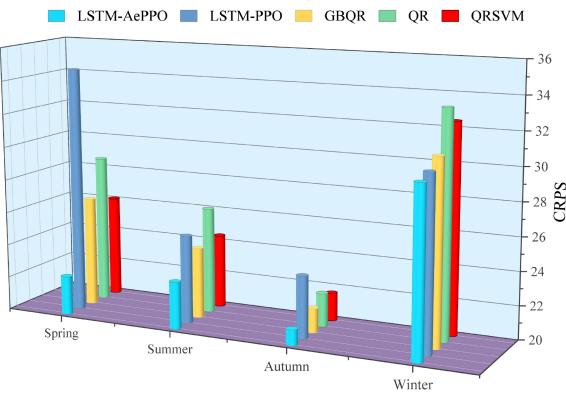


Fig. 8. The comparisons of CRPS in the case of JPL among the five mentioned algorithms.

is introduced to evaluate the accuracy of the whole forecast probabilistic distribution, and the lower CRPS value indicates a more accurate forecast probabilistic distribution. The comparisons of CRPS under Caltech and JPL cases are illustrated in Fig. 7 and Fig. 8. It can be seen from the two figures that our proposed LSTM-AePPO achieves the lowest CRPS values in all the cases, which demonstrates the accuracy of LSTM-AePPO could surpass other comparison algorithms. Based on the analysis of the Winkler, Pinball and CRPS values under different cases, it could be concluded that the LSTM-AePPO is more effective than QR, GBQR, QR-SVM and LSTM-PPO.

#### D. The Effectiveness of AePPO

To illustrate the effectiveness of the proposed adaptive exploration mechanism in LSTM-AePPO, the reward curves of PPO and AePPO during the 10000 training iterations are compared in the Caltech and JPL cases, as shown in Fig. 9 and Fig. 10. In Fig. 9, the reward of PPO increases during the first 3000 iterations and stays steady. Similarly, as shown in Fig. 10, the reward of PPO continuously raise from -20 to -7.5 in the first 1000 iterations then maintains at a relatively stable level.

*Because of applying the adaptive exploration mechanism, the AePPO would focus on exploration in the early stage, which*

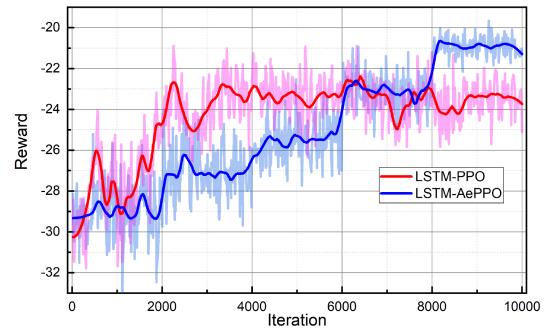


Fig. 9. The comparison between the reward of PPO and AePPO during the training in the case of Caltech.

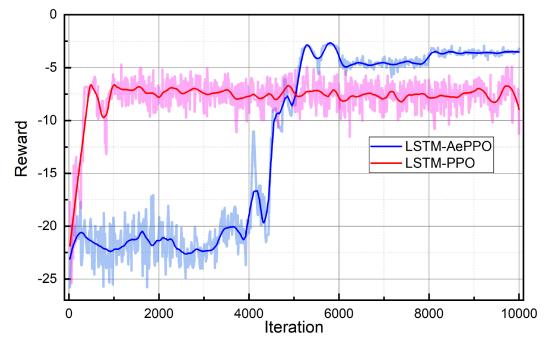


Fig. 10. The comparison between the reward of PPO and AePPO during the training in the case of JPL.

enriches the experience while causing the backwardness of the reward value. For instance, in the case of Caltech, the reward of PPO increases from around -30 to -23 during the early 3000 iterations, while the reward value of AePPO raises from -29 to -27. The same situation also appears in the case of JPL. The reward of PPO goes up from -25 to -7 in the early training, while the corresponding reward of AePPO hardly increases. However, the exploration of PPO would decrease due to the premature convergence of its actor, thus the diversity of its experience is not plentiful enough, which may cause the PPO falling into the local optima. As a result, the rewards of AePPO increase and surpass PPO after around the 8000th and the 5000th iteration in the case of Caltech and JPL, respectively.

Therefore, it can be seen that the adaptive exploration mechanism strengthen the accumulation of AePPO experiences in the early stage. Then, the AePPO would make full use of these experiences during the training and finally achieves a higher reward than PPO. In addition, the fluctuation of reward curves obtained by LSTM-AePPO are lower in the two cases. This illustrates that the proposed AePPO provides a more stable and superior performance than the original PPO.

## VII. CONCLUSION

This paper has proposed a reinforcement learning assisted deep learning probabilistic forecast framework for the charging power of EVCS, which contains a data transformer method and a probabilistic forecast algorithm, termed as LSTM-AePPO.

After being preprocessed by the data transformer, the charging session data is used to train the LSTM, which aims to obtain the mean value of the forecast distribution. Then, the variation of the LSTM cell state is modeled as a MDP and a reinforcement learning algorithm, AePPO, is applied to solve it. In this way, the variance of the forecast distribution can be provided by the AePPO. In addition, to balance the exploration and the exploitation, an adaptive exploration mechanism is further proposed to enrich the diversity of the experiences thus preventing the premature convergence of AePPO. Finally, the case studies are conducted based on the charging session data of Caltech and JPL. The experiments show the superior performance of the proposed LSTM-AePPO by comparing with QR, QRSVM, GBQR and LSTM-PPO on the CRPS, Winkler and Pinball metrics. Moreover, the comparison between the reward curves of PPO and AePPO indicates that the adaptive exploration mechanism is more effective in balancing the exploration and the exploitation of reinforcement learning.

Two challenges remain in our work, which are worth investigating in future work. First, the better point forecasting method should be investigated to achieve a higher prediction accuracy. Meanwhile, the method is also required to remain the parameters that contain the uncertainties of model and data simultaneously, such as the cell state in LSTM. The other is accelerating the training process of reinforcement learning algorithm, i.e., AePPO, while keeping the diversity of exploration and the convergence performance.

#### APPENDIX A THE CALCULATION OF CONTINUOUS RANKED PROBABILITY SCORE

Normally, the CRPS is defined as follows.

$$\text{CRPS}(\tilde{F}, y) = \int_{-\infty}^y \tilde{F}(x)^2 dx + \int_y^\infty (1 - \tilde{F}(x))^2 dx \quad (\text{A1})$$

where  $\tilde{F}(\cdot)$  is the predicted distribution obtained by the probabilistic forecasting model and  $y$  is the real value. Since the noise of our studied probabilistic forecast problem is assumed to follow the Gaussian distribution, as shown in (1), (4), and (A1) can be derived as follows.

$$\begin{aligned} \text{CRPS}(\tilde{F}, y) &= \delta \left( \frac{y - \mu}{\delta} \right) \left( 2\tilde{F} \left( \frac{y - \mu}{\delta} \right) - 1 \right) \\ &\quad + 2\tilde{f} \left( \frac{y - \mu}{\delta} \right) - \frac{1}{\sqrt{\pi}} \end{aligned} \quad (\text{A2})$$

where  $\mu$  and  $\delta$  are the mean and variance of the predicted distribution. The functions  $\tilde{F}(\cdot)$  and  $\tilde{f}(\cdot)$  are formulated as follows.

$$\tilde{f}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (\text{A3})$$

$$\tilde{F}(x) = \int_{-\infty}^x \tilde{f}(t) dt \quad (\text{A4})$$

#### APPENDIX B

#### THE PSEUDOCODE OF LSTM-AEPPo TRAINING

##### Algorithm 1: The Training of LSTM.

- 
- 1: **Input** The set of input features  $\{X_n\}_{n=1}^N$  and the target charging power set  $\{Y_n\}_{n=1}^N$ .
  - 2: **Output** A well trained LSTM model  $A(X_n, Y_n, c_n)$
  - 3: **Initialize**  $\eta_{lstm}$  (The learning rate of LSTM).
  - 4: **Initialize**  $N_{train}, N_{valid}, N_e$  (Number of samples in the training set, validation set and the number of training epochs)
  - 5: **for** epoch  $e = 1$  to  $N_e$  **do**
  - 6:   **Initialize**  $c_1$
  - 7:   Receive  $X_n$  and  $Y_n$
  - 8:   Execute the forward propagation of LSTM:
  - 9:    $f_{n+1} \leftarrow \sigma(W_f^n \cdot [Y_n, X_n] + b_f^n)$
  - 10:    $i_{n+1} \leftarrow \sigma(W_i^n \cdot [Y_n, X_n] + b_i^n)$
  - 11:    $\tilde{c}_{n+1} \leftarrow \tanh(W_c^n \cdot [Y_n, X_n] + b_c^n)$
  - 12:    $c_{n+1} \leftarrow f_{n+1} \otimes c_n + i_{n+1} \otimes \tilde{c}_{n+1}$
  - 13:    $o_{n+1} \leftarrow \sigma(W_o^n \cdot [Y_n, X_n] + b_o^n)$
  - 14:    $\hat{y}_{n+1} \leftarrow o_{n+1} \times \tanh(c_{n+1})$
  - 15:   Calculate the loss of LSTM:
  - 16:   Receive  $Y_{n+1}$
  - 17:    $\mathcal{L} \leftarrow \text{MSE}(\hat{y}_{n+1}, Y_{n+1})$
  - 18:   Calculate the derivatives of the learnable parameters of LSTM  $d\mathcal{L}/d\theta^n$ , where  $\theta^n$  represents  $W_f^n, W_i^n, W_c^n, W_o^n, b_f^n, b_i^n, b_c^n$  and  $b_o^n$ . The details are given in the Appendix D.
  - 19:   Update the parameters of the LSTM via Adam.
  - 20: **end for**
  - 21: **return**  $A(X_n, Y_n, c_n)$ .
- 

#### APPENDIX C COMPARATIVE METRICS IN EXPERIMENTS

To compare performances of the algorithms mentioned in this paper, the competitive metrics are needed to be used. The comprehensive evaluation of probabilistic prediction should consider its accuracy, variation, and reliability. Here, we adopt three metrics, i.e., CRPS, Winkler [23] and Pinball [30] to evaluate the three aspects, respectively. The definition of CRPS is given in (A2). As a famous metric, the Winkler is used to evaluate the variation of probabilistic distribution, and it is expressed as follows.

$$\text{Winkler} = \begin{cases} 2(L_p(\hat{y}_t) - y_t)/\alpha + \delta, & y_t < L_p(\hat{y}_t) \\ 2(y_t - U_p(\hat{y}_t))/\alpha + \delta, & y_t > U_p(\hat{y}_t) \\ \delta, & \text{otherwise} \end{cases} \quad (\text{C1})$$

where  $\hat{y}_t$  and  $y_t$  denote the predicted distribution and the real charging power at time  $t$ .  $L_p(\hat{y}_t)$  and  $U_p(\hat{y}_t)$  indicate the lower and upper quantiles at probability  $p$ .  $\alpha$  and  $\delta$  are two parameters of this metric, which are set as 0.1 and 1, respectively. Normally, the lower value of Winkler means the better variation.

In addition, considering the reliability of the predicted probabilistic distribution, Pinball is introduced as a metric, and its

**Algorithm 2:** The Training of AePPO.

---

```

1: Input The input features set  $\{X_n\}_{n=1}^N$  and target
   charging power set  $\{Y_n\}_{n=1}^N$ . The cell state produced
   by LSTM  $\{c_n\}_{n=1}^N$ .
2: Output A well trained actor  $\pi$ .
3: Initialize  $\eta_\pi, \eta_Q, \epsilon, \gamma, N_{train}, N_{valid}, N_e, N_{pso},$ 
    $N_{var}, e_1, e_2$ .
4: Initialize The actor and critic network  $\pi, V$ .
5: Initialize  $v_i^j, x_i^j$  and  $f_i^j$  of each particle  $i, i \in [1, N_{pso}]$ 
6: function fitness $x_i, b_{j,1}$ 
7:    $b_{j,2} = 1 - \sqrt{1 - (b_{j,1} - 1)^2}$ 
8:   Sample  $N_a$  action from  $\mathcal{N}(a_k, \text{mean}, x_i)$ 
9:    $r_k^{\text{mean}} \leftarrow \frac{1}{N_a} \sum_{j=1}^{N_a} r(s_k, a_k^j)$ 
10:   $r_k^{\text{var}} \leftarrow \frac{1}{N_a} \sum_{j=1}^{N_a} (r(s_k, a_k^j) - r_k^{\text{mean}})$ 
11:  return  $b_{j,1}r_k^{\text{mean}} + b_{j,2}r_k^{\text{var}}$ 
12: end function
13: for epoch  $e = 1$  to  $N_e$  do
14:   Collect the experience tuple  $\langle s_k, a_k, r_k, s_{k+1} \rangle$ :
15:   for  $k = 1$  to  $N_{train}$  do
16:      $s_k \leftarrow c_k$ 
17:      $a_k \leftarrow \pi(s_k)$ 
18:      $r_k$  is obtained by (A2)
19:      $s_{k+1} \leftarrow c_{k+1}$ 
20:   end for
21:   Compute the discounted rewards:
22:    $Q(s_k, a_k) \leftarrow r(s_k, a_k) + \sum_{i=1}^t \gamma^{t-i} r(s_{k-i}, a_{k-i})$ 
23:   Estimate the advantage:
24:    $A_{s,a}^{\pi_k} \leftarrow Q(s_k, a_k) - V^{\pi_k}(s_k)$ 
25:   Update the policy by minimizing the loss function:
26:    $\mathcal{L}^{\text{CLIP}} =$ 
       $\min\left(\frac{\pi_k(a|s)}{\pi_{k-1}(a|s)} A_{s,a}^{\pi_k}, \text{clip}\left(\frac{\pi_k(a|s)}{\pi_{k-1}(a|s)}, 1 - \epsilon, 1 + \epsilon\right) A_{s,a}^{\pi_k}\right)$ 
27:   The details are given in the Appendix D.
28:   Update the policy by minimizing the loss function:
29:    $\mathcal{L}^Q =$ 
       $\mathbb{E}_{s,a \sim \pi_k} [(\gamma V^{\pi_k}(s_{k+1}) + r(s_k, a_k) - V^{\pi_k}(s_k))^2]$ 
30:   Using PSO to obtain the adaptive exploration.
31: end for
32: return The well trained actor  $\pi$ 

```

---

mathematical formulation is expressed as follows.

$$\text{Pinball}_p = \begin{cases} (y_t - \hat{y}_{p,t})p, & \hat{y}_{p,t} < y_t \\ (\hat{y}_{p,t} - y_t)(1-p), & \hat{y}_{p,t} \geq y_t \end{cases} \quad (\text{C2})$$

where  $p$  stands for the probability of  $\hat{y}_{p,t} > y_t$  and  $\hat{y}_{p,t}$  indicates the predicted value at time  $t$ . The average Pinball value  $\text{Pinball} = \sum_p \text{Pinball}_p$  of the probabilistic prediction is used to represent its performance. Note that the lower Pinball stands for the better performance.

## APPENDIX D

THE DERIVATIVES OF LEARNABLE PARAMETERS IN  
LSTM-AEPPD

In this section, we provide the derivatives of the learnable parameters  $W_f, W_i, W_c, W_o, b_f, b_i, b_c, b_o$  and  $\theta$  of LSTM-AePPO.

Based on the (10)–(15), the LSTM takes  $X_t$  as its input and output  $y_{t+1}$  as its forecast. Its loss function is denoted as  $\mathcal{L}$ . The  $W_f$  is taken as an example to describe its derivative<sup>1</sup>.

$$\begin{aligned}
 \frac{d\mathcal{L}}{dW_f} &= \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial W_f^t} \\
 &= \left( \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial c^t} \right) + \frac{\partial c^{t+1}}{\partial W_f^t} \\
 &= \left( \left( \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial c^t} \right) + \frac{\partial c^{t+1}}{\partial c^t} \right) \frac{\partial c^t}{\partial f_t} \frac{\partial f_t}{\partial W_f} \\
 \end{aligned} \quad (\text{D1})$$

where  $\partial \mathcal{L}^{t+1}/\partial y^t$  is the derivative of error at  $t + 1$  with respect to  $y^t$ . In this paper, the loss function  $\mathcal{L}$  is mean square error, which could be formulated by

$$\mathcal{L}^t = \frac{1}{2}(y^t - \hat{y}^t)^2. \quad (\text{D2})$$

Then, the other derivatives are given in the following explicit formulas.

$$\frac{\partial \mathcal{L}^t}{\partial y^t} = y^t - \hat{y}^t \quad (\text{D3})$$

$$\frac{\partial y^t}{\partial c^t} = o_t(1 - tath(c_t)^2) \quad (\text{D4})$$

$$\frac{\partial c^{t+1}}{\partial c^t} = f_{t+1} \quad (\text{D5})$$

$$\frac{\partial c^t}{\partial f_t} = c^{t-1} f_t (1 - f_t) \quad (\text{D6})$$

$$\frac{\partial f_n}{\partial W_f} = [y_t, X_t] \quad (\text{D7})$$

Similarly, the derivative of  $W_i, W_c, W_o, b_f, b_i, b_c$  and  $b_o$  could be represented by the following equations.

$$\frac{d\mathcal{L}}{dW_i} = \left( \left( \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial c^t} \right) + \frac{\partial c^{t+1}}{\partial c^t} \right) \frac{\partial c^t}{\partial i_t} \frac{\partial i_t}{\partial W_i} \quad (\text{D8})$$

$$\frac{d\mathcal{L}}{dW_c} = \left( \left( \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial c^t} \right) + \frac{\partial c^{t+1}}{\partial c^t} \right) \frac{\partial c^t}{\partial \tilde{c}_t} \frac{\partial \tilde{c}_t}{\partial W_c} \quad (\text{D9})$$

$$\frac{d\mathcal{L}}{dW_o} = \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial o_t} \frac{\partial o_t}{\partial W_o} \quad (\text{D10})$$

$$\frac{d\mathcal{L}}{db_f} = \left( \left( \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial c^t} \right) + \frac{\partial c^{t+1}}{\partial c^t} \right) \frac{\partial c^t}{\partial f_t} \frac{\partial f_t}{\partial b_f} \quad (\text{D11})$$

<sup>1</sup>During the intermediate training of LSTM, the  $\frac{\partial \mathcal{L}^t}{\partial y^t}$  may be equal to 0 because the loss function has not calculated at time  $t$ . In this case, to avoid the gradient disappearance that caused by the zero of derivative, we use  $(\frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t})$  instead of  $\frac{\partial \mathcal{L}^t}{\partial y^t}$ .

$$\frac{d\mathcal{L}}{db_i} = \left( \left( \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial c^t} \right) + \frac{\partial c^{t+1}}{\partial c^t} \right) \frac{\partial c^t}{\partial i_t} \frac{\partial i_t}{\partial b_i} \quad (\text{D12})$$

$$\frac{d\mathcal{L}}{db_c} = \left( \left( \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial c^t} \right) + \frac{\partial c^{t+1}}{\partial c^t} \right) \frac{\partial c^t}{\partial \tilde{c}_t} \frac{\partial \tilde{c}_t}{\partial b_c} \quad (\text{D13})$$

$$\frac{d\mathcal{L}}{db_o} = \left( \frac{\partial \mathcal{L}^t}{\partial y^t} + \frac{\partial \mathcal{L}^{t+1}}{\partial y^t} \right) \frac{\partial y^t}{\partial o_t} \frac{\partial o_t}{\partial b_o} \quad (\text{D14})$$

where

$$\frac{\partial c^t}{\partial i_t} = \tilde{c}^t i_t (1 - i_t) \quad (\text{D15})$$

$$\frac{\partial c^t}{\partial \tilde{c}_t} = i_t \quad (\text{D16})$$

$$\frac{\partial y^t}{\partial o_t} = \text{tanh}(c_t) \quad (\text{D17})$$

$$\frac{\partial i_t}{\partial W_i} = \frac{\partial \tilde{c}_t}{\partial W_c} = \frac{\partial o_t}{\partial W_o} = [y_t, X_t] \quad (\text{D18})$$

$$\frac{\partial f_t}{\partial b_f} = \frac{\partial i_t}{\partial b_i} = \frac{\partial \tilde{c}_t}{\partial b_c} = \frac{\partial o_t}{\partial b_o} = 1 \quad (\text{D19})$$

Then, the learnable parameters  $\theta$  of AePPO is provided. To simplify the notation, we denote  $\pi_t(a|s)/\pi_{t-1}(a|s)$  as the probability ratio by  $r(a|s)$ . Then, the (20) could be rewritten as

$$\begin{aligned} \mathcal{L}^{\text{CLIP}} &= \\ \mathbb{E}_{s,a \sim \pi_t} &\left[ \min(r(a|s)A_{s,a}^{\pi_t}, \text{clip}(r(a|s), 1-\epsilon, 1+\epsilon)A_{s,a}^{\pi_t}) \right] \end{aligned} \quad (\text{D20})$$

The derivative of  $\theta$  could be derived from the above formula.

$$\frac{d\mathcal{L}^{\text{CLIP}}}{d\theta} = \mathbb{E}_{s,a \sim \pi_k} \left[ \frac{\partial \log \pi_t^\theta(a|s)}{\partial \theta} r(a|s) A_{s,a}^{\pi_t} \right] \quad (\text{D21})$$

where  $\pi_t^\theta$  indicates the *actor* at iteration  $t$  with parameter  $\theta$ . Note that the term  $r(a|s)A_{s,a}^{\pi_t}$  in the above formula is clipped into the range  $[1-\epsilon, 1+\epsilon]$  [31]. The  $\partial \log \pi_\theta(a|s)/\partial \theta$  is formulated as follows.

$$\begin{aligned} \frac{\partial \log \pi_\theta(a|s)}{\partial \theta} &= \\ &= \frac{1}{(a_{\text{var}}^{\text{AePPO}})^2} \left[ (a - a_{\text{mean}}^\theta) \left( \frac{(a - a_{\text{mean}}^\theta)^2 - (a_{\text{var}}^{\text{AePPO}})^2}{a_{\text{var}}^{\text{AePPO}}} \right) \right] \end{aligned} \quad (\text{D22})$$

where  $a_{\text{var}}^{\text{AePPO}}$  is obtained by the adaptive exploration mechanism rather than AePPO during the training, which is different with traditional PPO. Besides,  $a_{\text{mean}}^\theta$  is obtained by AePPO with learnable parameters  $\theta$ .

With the formulation of these derivative, the learnable parameters could be update through the following equation with learning rate  $\eta$ .

$$P^t = P^t + \Delta P^t \quad (\text{D23})$$

where  $P^t$  indicates the learnable parameters of LSTM-AePPO and the  $\Delta P^t$  is represented by

$$\Delta P^t = \Delta P^{t-1} + \eta \frac{\partial \mathcal{L}}{\partial P^t} \quad (\text{D24})$$

where  $\partial \mathcal{L}/\partial P^t$  denotes the derivative of  $P^t$ .

## REFERENCES

- [1] Z. Wei, Y. Li, and L. Cai, "Electric vehicle charging scheme for a park-and-charge system considering battery degradation costs," *IEEE Trans. Intell. Veh.*, vol. 3, no. 3, pp. 361–373, Sep. 2018.
- [2] L. Liu, F. Kong, X. Liu, Y. Peng, and Q. Wang, "A review on electric vehicles interacting with renewable energy in smart grid," *Renew. Sustain. Energy Rev.*, vol. 51, pp. 648–661, 2015.
- [3] Y. Shi, H. D. Tuan, A. V. Savkin, T. Q. Duong, and H. V. Poor, "Model predictive control for smart grids with multiple electric-vehicle charging stations," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2127–2136, Mar. 2019.
- [4] K. Chaudhari, N. K. Kandasamy, A. Krishnan, A. Ukil, and H. B. Gooi, "Agent-based aggregated behavior modeling for electric vehicle charging load," *IEEE Trans. Ind. Inform.*, vol. 15, no. 2, pp. 856–868, Feb. 2019.
- [5] S. Cheng, Z. Wei, D. Shang, Z. Zhao, and H. Chen, "Charging load prediction and distribution network reliability evaluation considering electric vehicles' spatial-temporal transfer randomness," *IEEE Access*, vol. 8, pp. 124084–124096, 2020.
- [6] L. Chen, F. Yang, Q. Xing, S. Wu, R. Wang, and J. Chen, "Spatial-temporal distribution prediction of charging load for electric vehicles based on dynamic traffic information," in *Proc. IEEE 4th Conf. Energy Internet Energy Syst. Integr.*, 2020, pp. 1269–1274.
- [7] Y. Zheng, Z. Shao, Y. Zhang, and L. Jian, "A systematic methodology for mid-and-long term electric vehicle charging load forecasting: The case study of Shenzhen, China," *Sustain. Cities Soc.*, vol. 56, 2020, Art. no. 102084.
- [8] H. J. Feng, L. C. Xi, Y. Z. Jun, Y. X. Ling, and H. Jun, "Review of electric vehicle charging demand forecasting based on multi-source data," in *Proc. IEEE Sustain. Power Energy Conf.*, 2020, pp. 139–146.
- [9] H. M. Louie, "Time-series modeling of aggregated electric vehicle charging station load," *Electric Power Compon. Syst.*, vol. 45, no. 14, pp. 1498–1511, 2017.
- [10] X. Zhang, "Short-term load forecasting for electric bus charging stations based on fuzzy clustering and least squares support vector machine optimized by wolf pack algorithm," *Energies*, vol. 11, no. 6, 2018, Art. no. 1449.
- [11] A. Almaghrebi, F. Aljuheshi, M. Rafaei, K. James, and M. Alahmad, "Data-driven charging demand prediction at public charging stations using supervised machine learning regression methods," *Energies*, vol. 13, no. 16, 2020, Art. no. 4231.
- [12] J. Zhu *et al.*, "Electric vehicle charging load forecasting: A comparative study of deep learning approaches," *Energies*, vol. 12, no. 14, 2019, Art. no. 2692.
- [13] Z. Li, Y. Li, Y. Liu, P. Wang, R. Lu, and H. B. Gooi, "Deep learning based densely connected network for load forecasting," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 2829–2840, Jul. 2021.
- [14] G. Guo, W. Yuan, Y. Lv, W. Liu, and J. Liu, "Traffic forecasting via dilated temporal convolution with peak-sensitive loss," *IEEE Intell. Transp. Syst. Mag.*, to be published, doi: [10.1109/MITTS.2021.3119869](https://doi.org/10.1109/MITTS.2021.3119869).
- [15] M. Xue, L. Wu, Q. P. Zhang, J. X. Lu, X. Mao, and Y. Pan, "Research on load forecasting of charging station based on XGBoost and LSTM model," *J. Phys., Conf. Ser.*, vol. 1757, no. 1, 2021, Art. no. 012145.
- [16] Y. Kim and S. Kim, "Forecasting charging demand of electric vehicles using time-series models," *Energies*, vol. 14, no. 5, 2021, Art. no. 1487.
- [17] L. Buzna *et al.*, "An ensemble methodology for hierarchical probabilistic electric vehicle load forecasting at regular charging stations," *Appl. Energy*, vol. 283, 2021, Art. no. 116337.
- [18] X. Zhang, K. W. Chan, H. Li, H. Wang, J. Qiu, and G. Wang, "Deep-learning-based probabilistic forecasting of electric vehicle charging load with a novel queuing model," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3157–3170, Jun. 2021.
- [19] L. Zhu and N. Laptev, "Deep and confident prediction for time series at uber," *Proc. IEEE Int. Conf. Data Mining Workshops*, 2017, pp. 103–110.

- [20] Y. Li, G. Hao, Y. Liu, Y. Yu, Z. Ni, and Y. Zhao, "Many-objective distribution network reconfiguration via deep reinforcement learning assisted optimization algorithm," *IEEE Trans. Power Del.*, to be published, doi: [10.1109/TPWRD.2021.3107534](https://doi.org/10.1109/TPWRD.2021.3107534).
- [21] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1033–1044, May 2014.
- [22] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1341–1356, Sep. 2011.
- [23] M. Sun, T. Zhang, Y. Wang, G. Strbac, and C. Kang, "Using Bayesian deep learning to capture uncertainty for residential net load forecasting," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 188–201, Jan. 2020.
- [24] Y. Chu and C. F. Coimbra, "Short-term probabilistic forecasts for direct normal irradiance," *Renew. Energy*, vol. 101, pp. 526–536, 2017.
- [25] S. Kachroudi, M. Grossard, and N. Abroug, "Predictive driving guidance of full electric vehicles using particle swarm optimization," *IEEE Trans. Veh. Technol.*, vol. 61, no. 9, pp. 3909–3919, Nov. 2012.
- [26] Z. J. Lee, T. Li, and S. H. Low, "ACN-data: Analysis and applications of an open EV charging dataset," in *Proc. 10th Int. Conf. Future Energy Syst.*, 2019, pp. 139–149.
- [27] Y. He, R. Liu, H. Li, S. Wang, and X. Lu, "Short-term power load probability density forecasting method using Kernel-based support vector quantile regression and copula theory," *Appl. Energy*, vol. 185, pp. 254–266, 2017.
- [28] T. Hong, P. Wang, and H. L. Willis, "A Naïve multiple linear regression benchmark for short term load forecasting," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2011, pp. 1–6.
- [29] Y. Wang, N. Zhang, Y. Tan, T. Hong, D. S. Kirschen, and C. Kang, "Combining probabilistic load forecasts," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3664–3674, Jul. 2019.
- [30] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic individual load forecasting using pinball loss guided LSTM," *Appl. Energy*, vol. 235, pp. 10–20, 2019.
- [31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.



**Yuanzheng Li** (Member IEEE) received the M.S. degree in electrical engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2011 and the Ph.D. degree in electrical engineering from the South China University of Technology, Guangzhou, China, in 2015. He is currently an Associate Professor with HUST. He has authored or coauthored several peer-reviewed papers in international journals. His research interests include electric vehicle, deep learning, reinforcement learning, smart grid, optimal power system/microgrid

scheduling and decision making, stochastic optimization considering large-scale integration of renewable energy into the power system and multi-objective optimization.



**Shangyang He** received the B.S. degree in automation from the Wuhan University of Technology, Wuhan, China, in 2020. He is currently working toward the M.S. degree from the Huazhong University of Science and Technology, Wuhan, China. His research interests include electric vehicle charging power forecast, deep learning, deep reinforcement learning, and graph neural network.



**Yang Li** (Senior Member IEEE) received the Ph.D. degree in electrical engineering from North China Electric Power University, Beijing, China, in 2014. He is currently an Associate Professor with the School of Electrical Engineering, Northeast Electric Power University, Jilin, China. He is also a China Scholarship Council-funded Postdoc with Argonne National Laboratory, Lemont, IL, USA. His research interests include power system stability and control, integrated energy systems, renewable energy integration, and smart grids.



computing, and big data.

**Leijiao Ge** (Senior Member IEEE) received the bachelor's degree in electrical engineering and its automation from Beihua University, Jilin, China, in 2006, and the master's degree in electrical engineering from the Hebei University of Technology, Tianjin, China, in 2009, and the Ph.D. degree in electrical engineering from Tianjin University, Tianjin, China, in 2016. He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University. His main research interests include situational awareness of smart distribution network, cloud



**Suhua Lou** received the B.E., M.Sc., and Ph.D. degrees in electrical engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1996, 2001 and 2005, respectively. She is currently a Professor with HUST. Her research interests include power system planning, energy economics, and renewable energy generation.



**Zhigang Zeng** (Fellow IEEE) received the Ph.D. degree in systems analysis and integration from the Huazhong University of Science and Technology, Wuhan, China, in 2003. He is currently a Professor with the School of Automation and the Key Laboratory of Image Processing and Intelligent Control of the Education Ministry of China, Huazhong University of Science and Technology. He has authored or coauthored more than 100 international journal articles. His research interests include theory of functional differential equations and differential equations with discontinuous right-hand sides and their applications to dynamics of neural networks, memristive systems, and control systems. Dr. Zeng was an Associate Editor for the *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS* from 2010 to 2011. He has been an Associate Editor for the *IEEE TRANSACTIONS ON CYBERNETICS* since 2014 and the *IEEE TRANSACTIONS ON FUZZY SYSTEMS* since 2016, and a Member of the Editorial Board of *Neural Networks* since 2012, *Cognitive Computation* since 2010, and *Applied Soft Computing* since 2013.