Advances in Applied Energy 10 (2023) 100142

Contents lists available at ScienceDirect

Advances in Applied Energy

journal homepage: www.elsevier.com/locate/adapen

An adaptive deep-learning load forecasting framework by integrating transformer and domain knowledge

Jiaxin Gao[a,b], Yuntian Chen[a,b,*], Wenbo Hu[c], Dongxiao Zhang[a,d,e,*]

[a] Eastern Institute for Advanced Study, Eastern Institute of Technology, Ningbo, Zhejiang, P. R. China
[b] School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai, P. R. China
[c] School of Computer and Information, Hefei University of Technology, Hefei, P. R. China
[d] Department of Mathematics and Theories, Peng Cheng Laboratory, Guangdong, P. R. China
[e] National Center for Applied Mathematics Shenzhen (NCAMS), Southern University of Science and Technology, Guangdong, P. R. China

ARTICLE INFO

Keywords:
Load forecasting
Deep-learning
Domain knowledge
Transfer learning
Online learning
Interpretability

ABSTRACT

Electrical energy is essential in today's society. Accurate electrical load forecasting is beneficial for better scheduling of electricity generation and saving electrical energy. In this paper, we propose an adaptive deep-learning load forecasting framework by integrating Transformer and domain knowledge (Adaptive-TgDLF). Adaptive-TgDLF introduces the deep-learning model Transformer and adaptive learning methods (including transfer learning for different locations and online learning for different time periods), which captures the long-term dependency of the load series, and is more appropriate for realistic scenarios with scarce samples and variable data distributions. Under the theory-guided framework, the electrical load is divided into dimensionless trends and local fluctuations. The dimensionless trends are considered as the inherent pattern of the load, and the local fluctuations are considered to be determined by the external driving forces. Adaptive learning can cope with the change of load in location and time, and can make full use of load data at different locations and times to train a more efficient model. Cross-validation experiments on different districts show that Adaptive-TgDLF is approximately 16% more accurate than the previous TgDLF model and saves more than half of the training time. Adaptive-TgDLF with 50% weather noise has the same accuracy as the previous TgDLF model without noise, which proves its robustness. We also preliminarily mine the interpretability of Transformer in Adaptive-TgDLF, which may provide future potential for better theory guidance. Furthermore, experiments demonstrate that transfer learning can accelerate convergence of the model in half the number of training epochs and achieve better performance, and online learning enables the model to achieve better results on the changing load.

1. Introduction

Electrical energy is one of the most important forms of energy globally, with a profound impact on both industrial output and human life. The field of energy planning is fundamentally reliant on short-term electrical load forecasting. With regard to supply planning, generation reserve, system security, dispatching scheduling, demand-side management, and other decision-making processes in power systems, short-term electrical load forecasting supports power system operators [1]. Accurate short-term electrical load forecasting is becoming increasingly essential in order to effectively schedule electricity generation and save energy [2]. Indeed, a study shows that an improvement of only 1% in load forecasting can save up to £10 million annually for the thermal British power system [3].

Researchers are increasingly interested in electrical load forecasting, and many forecasting models have been developed based on it. These models are separated into two categories: domain knowledge-based models; and data-driven models.

For the knowledge-based models, Rahman and Bhatnagar created an expert system for load forecasting [2]. Although this system possesses good interpretability, it struggles to simulate complicated nonlinear interactions between features. Hassan et al. proposed an approach of interval type-2 fuzzy logic systems (IT2FLS) for electrical load forecasting [4], and Ali et al. developed a fuzzy logic model for load forecasting based on the weather factors and historical load data [5]. In the above two models, fuzzy rules (a kind of domain knowledge) play a key role. Both models have achieved good results, but the correctness of those fuzzy rules has not been fully proven, and the models have not fully mined the potential rules in the data.

With the development of artificial intelligence, data-driven models are widely used in many disciplines [6–9]. In the field of load forecasting, Park et al. used ANN to predict the future load [10]. However,

* Corresponding authors.
E-mail addresses: ychen@eias.ac.cn, cyt_cn@126.com (Y. Chen).

https://doi.org/10.1016/j.adapen.2023.100142
Received 27 February 2023; Received in revised form 3 May 2023; Accepted 7 May 2023
Available online 8 May 2023

their model is too simple, and it did not utilize temporal information to aid training. Bedi and Toshniwal employed long short-term memory (LSTM) for electrical load forecasting [11,12]. Shi et al. proposed a novel pooling-based deep recurrent neural network (PDRNN) for load forecasting to address the over-fitting issues of deep learning [13]. Ouyang et al. combined DBN with the Copula Model to mitigate the challenge of high variability and volatility of power grid network systems [14]. Dai et al. improved the support vector machine (SVM) for load forecasting [15]. Jurasovic et al. utilized Transformer for day-ahead load forecasting and achieved good results [16,17]. These data-driven models have high requirements on the amount of data, and without the assistance of domain knowledge, the models may be trapped in a local minimum and fail to achieve the highest accuracy during training.

Pure domain knowledge-based models and pure data-driven models are typically insufficient in handling complex problems and, as a consequence, some research has attempted to combine domain knowledge with data-driven models. Domain knowledge, on the other hand, is frequently employed mainly for feature engineering and has yet to be fully integrated with deep-learning algorithms [18,19]. In this case, domain knowledge is often underutilized. A theory-guided framework was proposed to address this problem. Domain knowledge and data-driven algorithms can be fully integrated under this framework, which is based on the usage of first-principle models and empirical models as the reference and basis for model prediction. This framework has been successfully applied to many problems. For example, Wang et al. and He et al. used theory-guided neural networks (TgNNs) in the field of hydrology and achieved good results [20,21]. Raissi et al. utilized the physics-informed neural network (PINN), which is essentially a type of theory-guided neural network (TgNN), for computational fluid dynamics and achieved good performance [22]. Karpatne et al. proposed theory-guided data science (TGDS), and presented five ways to integrate scientific knowledge and data science [23]. They also proposed the physics-guided neural network (PGNN) for lake temperature modeling [24]. He et al. developed the theory-guided full convolutional neural network (TgFCNN) to solve inverse problems in subsurface contaminant transport [25]. Li et al. proposed a TgNN as a prediction model for oil/water phase flow [26]. Chen et al. developed a kind of hard constraint model under the theory-guided framework to ensure that the model outputs obey known governing equations [27]. Previous theory-guided methods mostly used control equations, but in the field of electrical load forecasting, the physical process is too complicated, and there is still a lack of effective control equations. As a result, referring to the idea of theory-guided methods, the theory-guided deep-learning framework for load forecasting via ensemble long short-term memory (TgDLF) [28] was advanced.

TgDLF predicted local fluctuations using EnLSTM [29,30] and domain knowledge to obtain dimensionless trends. TgDLF outperformed knowledge-based and data-driven models. However, TgDLF has not yet solved practical load forecasting challenges including insufficient historical load data and load distribution change. Historical load data are insufficient for new target districts. Economic development may also alter load distribution in the same district.

The model's expressive capacity and historical load data size determine the load forecasting model's capacity. Using models with more expressive capacity improves feature extraction from historical load series, and adding training load data improves the model's understanding of load series data distribution. EnLSTM in TgDLF outperforms LSTM, but it inherits LSTM's inability to capture long-term dependencies, which limits its expression and forecasting capacity. TgDLF also ignores the issue of load distribution change and insufficient historical load data. In districts with limited historical load data, such as newly established districts or those with inadequate load data collectors, its functionality may be suboptimal. Insufficient training data can hinder the model's load data distribution learning and forecasting effectiveness.

Considering the aforementioned problems, a salient question is: are there more efficient time-series prediction models that can replace the EnLSTM model, and can solve the pragmatic problems of insufficient historical load data and load distribution change?

We present Adaptive-TgDLF, an improved version of TgDLF. Adaptive-TgDLF substitutes the EnLSTM model with a more expressive and efficient Transformer [16] model to improve accuracy and save training time. Adaptive-TgDLF addresses load distribution change and insufficient historical load data with adaptive learning methods. Transfer learning [31–33] and online learning [34,35] increase model performance in different districts and in the same district at different times. The expensive expense of electrical load collection and data privacy sometimes result in insufficient historical electrical load [36], which affects model performance. Transfer learning helps model train faster and better in Adaptive-TgDLF. District load data distribution may also change [35], and online learning in Adaptive-TgDLF helps adapt to this change and improve load forecasting results.

Adaptive-TgDLF outperforms TgDLF for time-series forecasting when trained with the same amount of data because Transformer is better at capturing long-term dependencies than EnLSTM. Adaptive-TgDLF also addresses insufficient history data and changing load distributions by using the model Transformer's expressive capacity and adaptive learning to add training.

The contribution of this study is four-fold:

- This study proposes a stronger and more robust model for short-term load forecasting (one day ahead hourly load forecasting), which is 16.3% lower than the previous efficient model TgDLF in MSE.
- This study proposes adaptive learning to solve the problem of insufficient historical load data and load distribution change, in which transfer learning and online learning improve the generalization ability of model in space and time, respectively.
- This study reduces half of the training time to train an electricity load forecasting model, compared with TgDLF.
- This study preliminarily mines interpretability of the deep-learning model for load forecasting.

2. Methodology

In this work, adaptive deep-learning load forecasting (Adaptive-TgDLF) is used to predict the load ratio, and the real load can be recovered with the load ratio and historical load. In this section, the theory-guided framework is introduced first, followed by a demonstration of the deep-learning model Transformer and two adaptive learning methods: transfer learning; and online learning. Finally, we show how to combine the theory-guided framework with Transformer and adaptive learning for load forecasting.

2.1. Theory-guided framework

The theory-guided framework refers to using knowledge and theory to guide the training of deep-learning models. It combines the generality of human knowledge with the efficiency of data-driven models, as shown in Fig. 1. Research and experiments show that with the assistance of human knowledge, neural networks can avoid many detours to speed-up convergence, and often achieve higher accuracy. In this study, the theory-guided framework is used for load forecasting.

Concretely, the electrical load data are a continuous series, and we can therefore convert the origin load series into the load ratio series, as shown in Eq. (1). The functions and advantages of ratio conversion are shown in Appendix A. This conversion is also crucial for transfer learning.

The load ratio can then be further decomposed into the dimensionless trend and local fluctuation. The two parts decomposed correspond to the internal mode of the district (unchanged in a short time), and the external driving force affected by weather or other factors. When performing adaptive learning, the dimensional trends can be directly transferred between different districts or time periods, while the models are

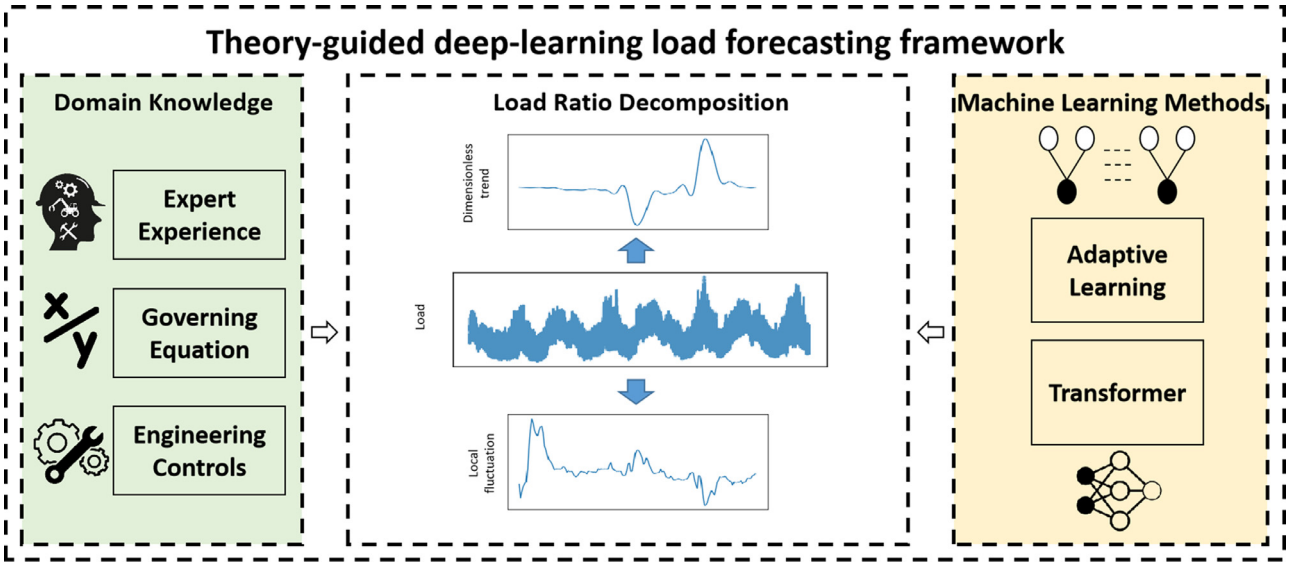

Fig. 1. Theory-guided framework.

re-trained with the local fluctuations. The dimensionless trend can be obtained based on the physical mechanism and domain knowledge, and deep-learning models can focus more on predicting local fluctuation. Subsequent experiments prove that this decomposition can greatly improve prediction performance. This process can be expressed by Eq. (1):

$$\begin{aligned} L_{t+1} &= Ratio_{t+1} * L_t \\ &= f(x,t) * L_t \\ &= \left(f_1(x,t) + f_2(x,t)\right) * L_t \\ f_1(x,t) &= DT_{t+1} \\ f_2(x,t) &= \delta_{t+1} \end{aligned} \tag{1}$$

where L_{t+1} and L_t represent the load at time $t + 1$ and t, respectively; $Ratio_{t+1}$ represents the load ratio at time $t + 1$; $f_1(x,t)$ is the dimensionless trend (DT); and $f_2(x,t)$ is the local fluctuation (δ). A more detailed description of the decomposition process is introduced in the previous study TgDLF [28].

Furthermore, the theory-guided framework also emphasizes the importance of weather factors and calendar factors for electrical load forecasting [37,38]. Under the theory-guided framework, instead of training many sub-models specifically to model weather and calendar factors [12], we utilize a unified model which fully considers a range of influencing elements, and is suitable for a variety of seasons and weather.

It should be noted that although "theory-guided" is an important part of our method, it is not the core contribution of this study.

2.2. Transformer

Researchers are increasingly using Transformer in machine-learning tasks like natural language processing (NLP), computer vision (CV), recommendation systems, mathematics, and more [39–41]. Bert, GPT-3, DALL-E, Codex [42–45], and more potential applications also use Transformer. Transformer is useful in time-series processing because it captures long-term feature dependency [46].

Transformer is made up of two parts: an encoder and a decoder. The encoder is responsible for high-level feature extraction. It is a stack of encoder blocks, and each block contains a multi-head attention module and a position-wise feed-forward network (FFN). The decoder is utilized for the final prediction. It is a stack of decoder blocks, each of which contains two multi-head attention modules and an FFN. In addition, a residual connection is adopted for constructing a deeper model [16,47].

Attention, a significant component of Transformer, enables features to interact, generating expressive features. It consists of three elements: query, key, and value. All features are interconnected with attention, capturing long-term dependencies (dependency information spanning up to 96 h, as input is 4 d of historical data) challenging for RNN and LSTM.

Specifically, an attention function is defined as a mapping function, which computes the inner product between the query and key vectors as the attention scores. The scaled dot-product attention used by Transformer is as follows:

$$\text{Attention}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \text{softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{D_K}}}\right)\mathrm{V} = \mathrm{AV} \tag{2}$$

where Q is queries; K is keys; V is values; Q, K, and V are generally obtained by applying some transformations to the original input, while they can also be obtained through external input; $\mathrm{D_k}$ represents the dimensions of keys; and A is often called an attention matrix, which can visualize the relationships between features. This attention matrix also provides interpretability for Transformer [48,49], which constitutes another advantage over conventional models for time-series data, such as the original LSTM. Some improved RNN-based models combined with the attention mechanism also have interpretability [50].

Transformer usually adopts multi-head attention, instead of using a single attention function. The definition of multi-head attention is:

$$\begin{aligned} &\text{MultiHeadAttn}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \text{Concat}\left(\text{head}_1,\ldots,\text{head}_\mathrm{H}\right)\mathrm{W}^{\mathrm{O}} \\ &\text{where head}_\mathrm{i} = \text{Attention}\left(\mathrm{QW}_\mathrm{i}^{\mathrm{Q}},\mathrm{KW}_\mathrm{i}^{\mathrm{K}},\mathrm{VW}_\mathrm{i}^{\mathrm{V}}\right) \end{aligned} \tag{3}$$

where $\mathrm{W_i^Q}$, $\mathrm{W_i^K}$, and $\mathrm{W_i^V}$ are parameter matrices, projecting Q, K, and V to different representation subspaces, respectively. Studies have demonstrated that multi-heads can learn different dimensions of information [16,51], so that the performance of Transformer that uses multi-head attention is usually superior.

Transformer can also better utilize the parallel computing power of GPUs [52], and thus it is usually more efficient than serial input models, such as LSTM.

2.3. Transfer learning

Transfer learning is an important machine-learning method to solve the fundamental problem of insufficient training data. The concise definition of transfer learning is: given a source data D_S and learning task T_S, a target data D_T and learning task T_T, transfer learning is committed to using the knowledge in D_S and T_S to help improve the learning of target prediction function $f_T(\cdot)$ in D_T.

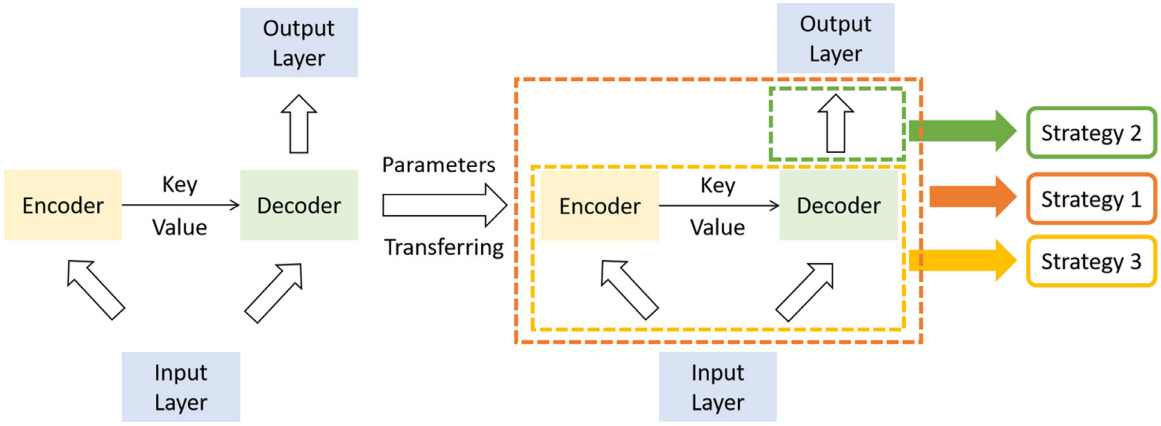

Fig. 2. Three strategies of transfer learning in Adaptive-TgDLF.

The premise of transfer learning is that the goals of the two tasks are the same or related, and the data used by the two tasks are similar or related [31]. In our application, our task in each district is load forecasting, and the load data in different districts are similar in data structure, and thus transfer learning can be used between different districts. More specifically, the transfer learning method used here belongs to transductive transfer learning [31].

Transfer learning uses load data from one district as the source data and load data from another district as the target data to improve deep-learning model performance. We first extract the dimensionless trend from the source data, and then train a model for the local fluctuation of the source data, which we refer to as the "source model". For transfer learning, the dimensionless trend is applied to the target data directly, and the source model is applied to the local fluctuations of the target data. Transformer consists of two parts: feature extraction (encoder, decoder, and embedding layers) and prediction (last fully connected layer). The former extracts original input features, and the latter predicts final results.

Three strategies are adopted for transfer learning while re-training the source model on the target data: (1) update all weights of the model while re-training; (2) fix the weights of the encoder, decoder, and embedding layers, and only update the weights of the last fully connected layer while re-training; and (3) fix the weights of the last fully connected layer, and only update the weights of the encoder, decoder, and embedding layers. These three strategies are illustrated in Fig. 2. We use these three strategies to test whether the feature extraction part transfer is more effective, or the prediction part transfer is more effective, or the entire model needs to be fine-tuned.

2.4. Online learning

Online learning is an important machine-learning method to adapt to changes in data distribution. Online learning is opposite to offline learning. Traditional offline learning trains a model from the entire historical data at once and deploys it for inference without updating [34]. Online learning is mostly used to overcome the data distribution change problem and make the model more suited for small sample scenarios. Online learning improves the model for two reasons: first, it expands the model's training set; second, the internal mode of test data may change over time, and online learning can modify the model to ensure that the model's prediction results don't gradually deviate from the actual situation during inference. Online learning can update the model to the newest load distribution in each district in our application. More specifically, the online learning method used here belongs to the optimization-based online learning method [35].

In online learning, the early load data of a district are chosen as the historical data, and the late load data of the district are chosen as the latest online data. We first extract the dimensionless trend from the historical data, and then train a model for the local fluctuation of the historical data, which we refer to as the "offline model". For online learning, the dimensionless trend from the historical data is applied to the latest online data directly; in this way, the local fluctuation of the online data can be acquired. Then, the offline model is re-trained with the local fluctuation of the online data to obtain the online model. The process of online learning is shown in Fig. 3.

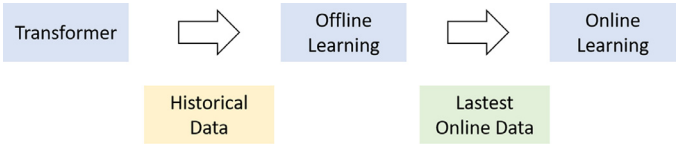

Fig. 3. Process of online learning.

2.5. Adaptive-TgDLF

In this study, based on Transformer's expressive representation capability and the effectiveness of adaptive learning (including transfer learning and online learning), we propose adaptive deep-learning load forecasting (Adaptive-TgDLF), which combines the theory-guided framework with Transformer and adaptive learning.

In this study, we first extract the weekly average trend of the load ratio from the training data, a low pass filter is then applied to smoothing the trend, and the filtered trend is used as the dimensionless trend. Regarding local fluctuation, Transformer is used to predict it. The structure of the model and the data flow inside of the model are illustrated in Fig. 4. The input of Transformer includes historical load, weather factors and calendar factors, and the output of Transformer if the prediction of load. The detailed description of the Transformer's input and output is given in Appendix B.

The whole process of Adaptive-TgDLF is shown in Fig. 5. The load ratio is acquired by adding the dimensionless trend and local fluctuation together, and then the load ratio can be restored to the real load with the real load from the previous day, as shown in Eq. (1). Adaptive learning methods are also effectively integrated into Adaptive-TgDLF.

3. Experiment

3.1. Data description and experiment setting

In this study, we take the electrical load data from 2008.01.01 to 2011.09.23 of 12 districts in Beijing, China, as a study case. The 12 districts are Chaoyang (CY), Haidian (HD), Fengtai (FT), Shijingshan (SJS), Pinggu (PG), Yizhuang (YZ), Changping (CP), Mentougou (MTG),

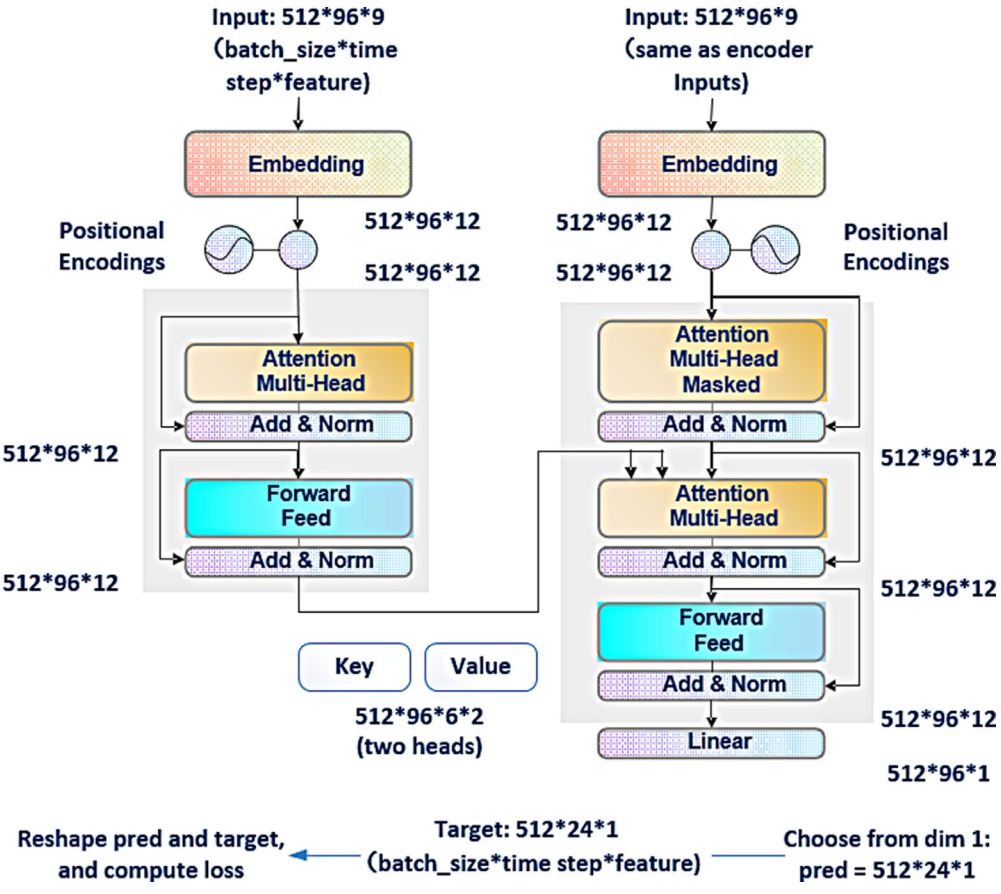

Fig. 4. Structure of Transformer.

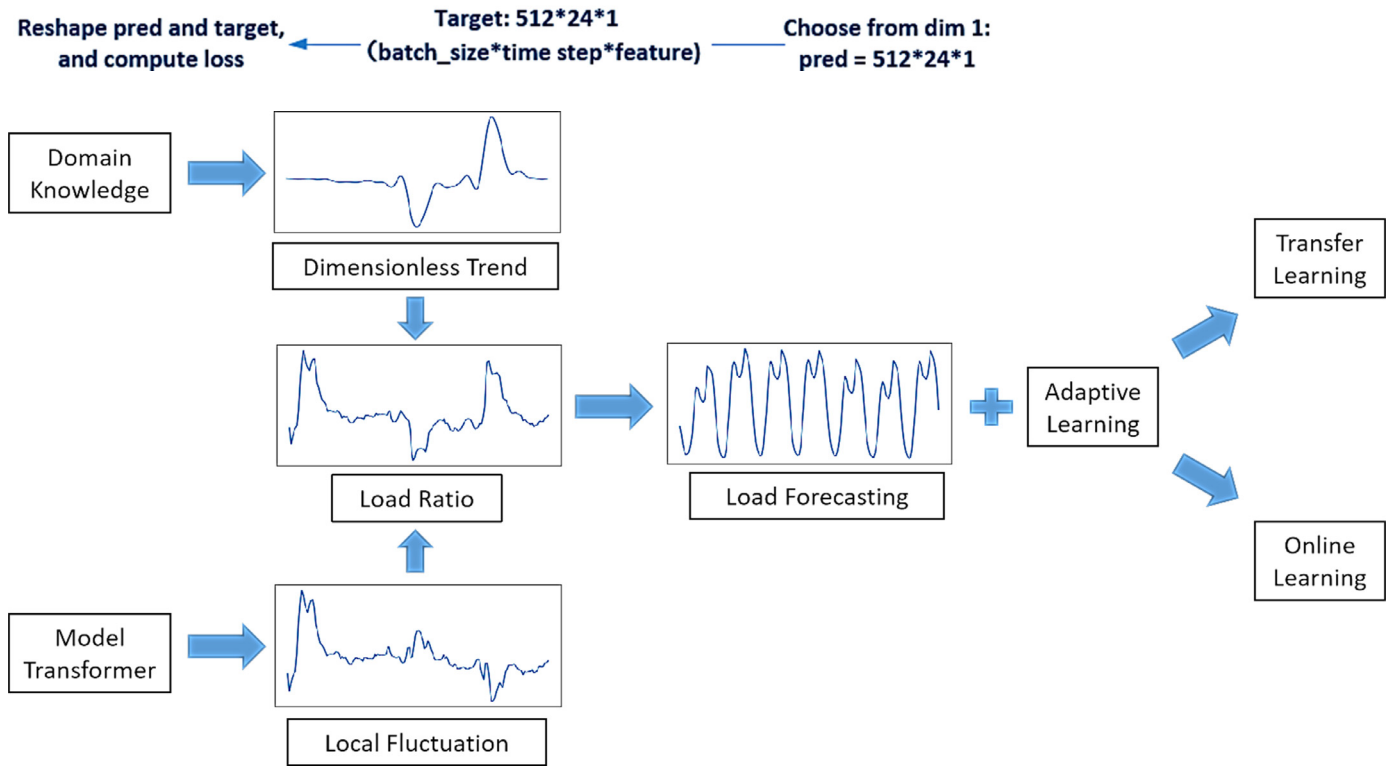

Fig. 5. Flowchart of Adaptive-TgDLF.

Fangshan (FS), Daxing (DX), Miyun (MY), and Shunyi (SY). The location relationship of the 12 districts is illustrated in Fig. 6. The load forecasting in our study is for different districts. It can obtain a comprehensive result of multiple sectors, not only for residential sectors, industrial sectors, business sectors, etc. These 12 districts can be divided into three groups according to correlation analysis: east Beijing (yellow); central Beijing (blue); and west Beijing (red). Since the load data and weather data are sampled per hour, there are 392,256 data in total.

The meteorological data in the 12 districts are also applied to load forecasting. The meteorological data include temperature, humidity, wind speed, and precipitation rate, which are four factors considered to be crucial for load forecasting.

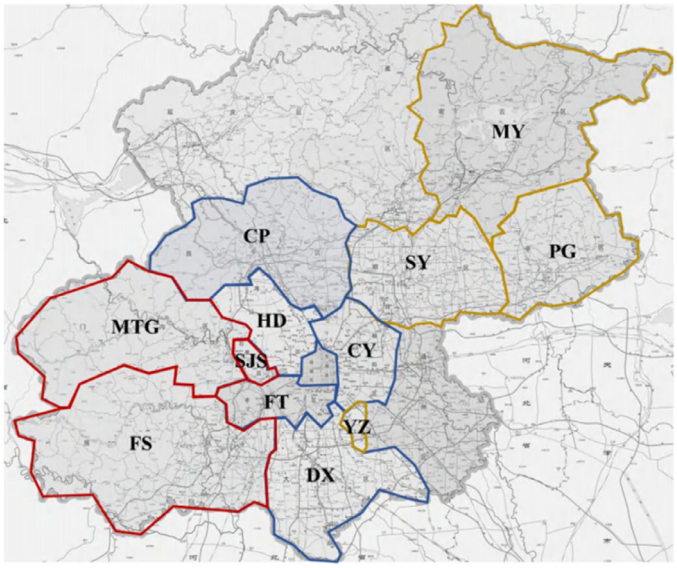

Fig. 6. Map of the 12 research districts in Beijing [28].

Since both load data and weather data have a small proportion of missing values or outliers, preprocessing of the data is required. For the load data, two data preprocessing methods are mainly used. First, linear interpolation is used for missing value filling, as the effect of polynomial interpolation is not significantly better than linear according to the experiment. Second, the outliers are detected according to their neighborhood after filling the missing values. Since the load data are relatively stable in the short-term, if the difference exceeds seven times the current median of the neighborhood (3 d of data points around the point), the data are regarded as an outlier, and thus we need to remove these outliers and use linear interpolation to fill in these values. For the meteorological data, the duplicated meteorological data are deleted at first, and missing values in precipitation rate are directly set to zero, because some non-rainfall periods are recorded as missing values in practice, according to our knowledge. Furthermore, it is found that the percentage of missing precipitation rate data is approximately 10% to 30%. This is because Beijing is dry in winter, and there is very little precipitation.

When using weather data, there are also two options: use historical weather data (W_t); and use weather data for the next day (W_{t+1}). Since the real weather data for the next day cannot be obtained in practice, the weather data for the next day can be replaced by weather forecast data for the next day (W'_{t+1}). Weather forecast data are simulated with real weather data and normally-distributed noise, as the weather forecast is always different from the actual weather. It is worth noting that the training set and test set contain real weather data, but they do not contain weather forecast data, and thus weather forecast data can be simulated in this way. When predicting the load for the next day in a real scenario, the real weather forecast data for the next day can be easily collected, and thus the real weather forecast data can be used. The formula for adding random noise to weather data is as follows:

$$W'_{t+1} = W_{t+1} * (1 + Noise * Proportion) \tag{4}$$

where W'_{t+1} and W_{t+1} represent the weather forecast data and real weather data at time $t + 1$, respectively; *Noise* represents normally-distributed noise; and *Proportion* represents the proportion of added noise.

The results in the TgDLF show that the use of weather forecast data is significantly better than the use of historical weather data [28], which is logical. Adaptive-TgDLF also uses weather forecast data, and the proportion of added noise to real weather data is 5%.

In addition, we find that the load always changes greatly on Saturday and Monday, and thus two flags (variables with a value of 0 or 1) are added to the input to indicate whether the day is Monday or whether it is Saturday. Essentially, these two flags represent the switch of weekday and weekend. Moreover, it is well known that people's electricity consumption habits on weekends are not the same as those during weekdays, and people consume more electricity than usual in summer; consequently, two more flags are added to indicate whether the day is on a weekend and whether it is in the summer. Finally, the input has nine dimensions: one load ratio; four weather factors; and four calendar factors.

In Adaptive-TgDLF, a moving window method is used to generate the samples for training and testing. Four days (4 d) of historical data are utilized to predict the load of 1 d in the future. The initial input window contains the first 4 d of historical load data, weather factors, and calendar factors, and the initial output window contains the load data for the fifth day. In this way, the first sample is generated. Each time that the input window slides forward 1 d, the output window also slides forward 1 d, and more samples are generated. Each region can contribute 1355 samples after sliding the load data, weather factors, and calendar factors.

3.2. Load forecasting experiments

In load forecasting experiments, to measure the performances of the models more objectively and make better use of the data, we adopt a four-fold cross-validation method [53], in which the 12 districts are divided into four folds, and each experiment takes three folds as training data and one fold as test data. We conduct four experiments in total, so that the performances of the model in each district can be acquired.

In experiments, mean square error (MSE) metrics are used to evaluate models. The results of cross-validation are shown in Table 1. Four baselines are selected: ARMA and ARIMA [54,55] are the baselines as traditional statistical models; LSTM is the baseline as a classical time-series deep-learning model; and the baseline DT represents a model that predicts the load only based on the dimensionless trend, and does not utilize machine-learning models to predict local fluctuations. In addition, we list the results of TgDLF, which is the previous theory-guided model. To demonstrate the importance of domain knowledge, the results of TgDLF and Adaptive-TgDLF without the assistance of dimensionless trend (EnLSTM and Transformer, respectively) are also listed in Table 1.

It is shown in Table 1 that Adaptive-TgDLF has an obvious advantage in each district. The MSE of Adaptive-TgDLF is 86.1%, 52.3%, and 35.4% lower than the MSE of ARMA, ARIMA, and LSTM, respectively. Furthermore, the MSE of Adaptive-TgDLF is 44.0% lower than the MSE of DT, which proves that only using dimensionless trends is insufficient. Moreover, Adaptive-TgDLF is 16.3% lower than its previous version TgDLF in MSE. The results of root mean square error (RMSE) metrics and mean absolute percentage error (MAPE) metrics are also listed in Appendix C to ensure that the comparisons are reliable. We used a GeForce GTX 1080ti GPU to complete the training and evaluate the training time of TgDLF and Adaptive-TgDLF.

Thanks to the training efficiency of Transformer, Adaptive-TgDLF saves more than half of the training time compared to TgDLF, as shown in Table 2. In future smart energy systems, fast real-time responses to higher time resolution load and real-time model optimization are very important, and the training speed of the model will become crucial [56,57].

3.3. Visualization, interpretability and robustness of Adaptive-TgDLF

In this subsection, we first visually show the prediction performance of Adaptive-TgDLF, and then analyze the interpretability of Transformer in Adaptive-TgDLF. Finally, we test the robustness of Adaptive-TgDLF against weather noise [58].

The predicted outcomes of Adaptive-TgDLF in the Fengtai district are used as an example in Fig. 7 to demonstrate the model performance

Table 1
Prediction MSE of different models.

	ARMA	ARIMA	LSTM	DT	EnLSTM	TgDLF	Transformer	Adaptive-TgDLF
PG	0.260	0.113	0.102	0.121	0.095	0.077	0.071	**0.061**
SJS	0.409	0.115	0.115	0.116	0.119	0.106	0.102	**0.097**
CY	0.071	0.062	0.052	0.065	0.046	0.032	0.027	**0.023**
YZ	0.160	0.121	0.091	0.126	0.089	0.080	0.068	**0.056**
MTG	2.296	0.324	0.111	0.123	0.120	0.107	0.105	**0.096**
FT	0.097	0.070	0.058	0.064	0.053	0.030	0.029	**0.028**
MY	0.188	0.083	0.072	0.078	0.065	0.049	0.050	**0.044**
FS	0.363	0.102	0.093	0.096	0.091	0.075	0.079	**0.071**
CP	0.144	0.069	0.059	0.056	0.053	0.040	0.038	**0.035**
SY	0.174	0.091	0.078	0.100	0.070	0.055	0.048	**0.041**
HD	0.081	0.078	0.064	0.081	0.053	0.042	0.038	**0.031**
DX	0.151	0.054	0.059	0.062	0.051	0.035	0.033	**0.028**
AVG	0.366	0.107	0.079	0.091	0.075	0.061	0.057	**0.051**

Table 2
Training time of TgDLF and Adaptive-TgDLF.

	AVG	Fold 0	Fold 1	Fold 2	Fold 3
TgDLF	769 s	762 s	758 s	774 s	780 s
Adaptive-TgDLF	**344 s**	**348 s**	**328 s**	**355 s**	**345 s**

in further detail. The electrical load (including local fluctuation, dimensionless trend, load ratio, and real load) is represented by the ordinate, and time is shown by the abscissa. We choose the time period from 2008.08.08 to 2008.09.17, during which the Olympic and Paralympic Games were held in Beijing. The red lines represent the prediction results of Adaptive-TgDLF; whereas, the black lines represent the actual value. It can be seen that the prediction of the local fluctuation by the model Transformer is good in most cases, but in some places with large changes, the prediction still has a certain bias. However, with the addition of human knowledge (dimensionless trend), this bias is reduced. We finally restore the load ratio back to the real load. Fig. 7 intuitively shows that the predicted load of Adaptive-TgDLF and the actual load are consistent.

The Transformer in Adaptive-TgDLF not only possesses high expressiveness, but also has good interpretability. The interpretability of Transformer is mainly reflected through attention [48,49], and it is possible to observe what Transformer has learned by visualizing the attention matrix.

Several typical attention matrices of Transformer and their corresponding samples are presented in Fig. 8. The three figures above are attention matrices, and the length of the horizontal and vertical coordinates of each attention matrix is 96, representing the 96 h of input. Each point in the attention matrix can be understood as a combined feature point formed by the interaction of the original features: for example, the point (15, 20) can be understood as the combined feature point formed by the interaction of the original feature of the 15th hour and the 20th hour. Overall, the larger is the value of this point, the greater is the influence that it has on the final prediction of the model Transformer. In the figure of attention matrix, the depth of the color can be used to represent the value of the point. Overall, the brighter is the color, the larger is the value in the attention matrix, and it can also be proven that the combined feature represented by the point is more important for the prediction of the model. The three figures below are samples corresponding to the attention matrices. The abscissa of the

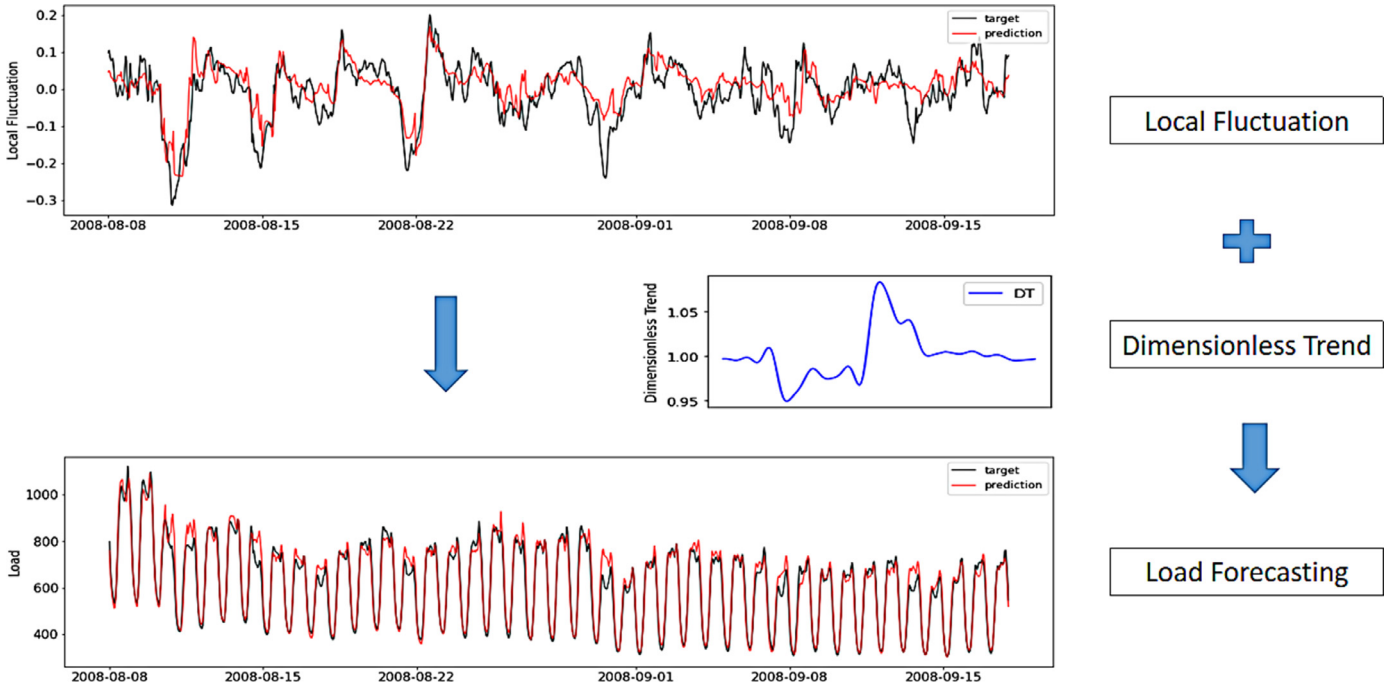

Fig. 7. Load forecasting of Adaptive-TgDLF in the Fengtai district.

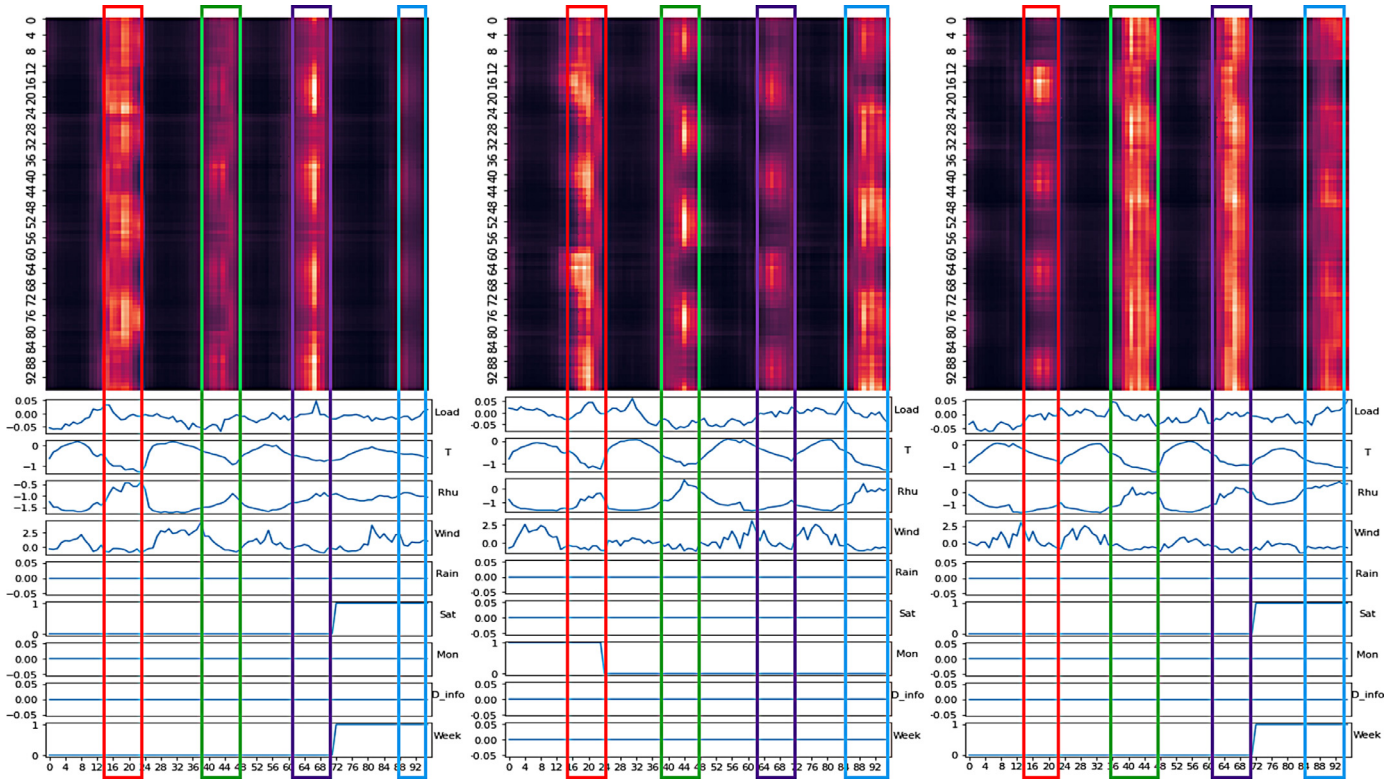

Fig. 8. Several typical attention matrices of Transformer.

figure also represents the input of 96 h The figure can be divided into nine subfigures, with each subfigure representing a feature of the input (load, temperature, humidity, wind speed, precipitation rate, whether it is Saturday, whether it is Monday, calendar effect, and whether it is a weekend, from top to bottom), and the ordinate of each subfigure represents the value of the feature.

These three attention matrices are band-shaped, and the highlighted parts are concentrated from 3:00 pm to 11:00 pm every day. In fact, we found that the attention matrices corresponding to most samples are band-shaped, and the highlighted time periods are basically the same. Since the local fluctuations to be predicted represent uncertainty, we suspect that this is due to the higher uncertainty of people's activity between 3:00 pm and 11:00 pm (people are generally at work before 3:00 pm and resting after 11:00 pm, and the uncertainty of people's activities within these two time periods will be much smaller). Since the time period from 3:00 pm to 11:00 pm can provide more information for the model prediction, the model will pay more attention to this time period of the day. The evolving of the attention matrices of a sample in the training process is provided in Appendix D. We believe that the samples corresponding to these attention matrices contain very complex nonlinear mapping relationships, and we will analyze them in the future. It can be seen that the Transformer model has achieved good prediction results based on these outstanding understandings of the samples. The interpretability of Transformer not only shows how the model works, but also provides potential for better theory guidance. For instance, it is found that the attention matrices in different districts are highly similar, suggesting that people in different districts have certain similar behavior patterns.

It is well known that the model's prediction accuracy can be significantly increased by using accurate weather forecast data [30]. Since weather forecast data cannot be absolutely accurate, we also conduct an experiment to assess the effect of noisy weather forecast data on load forecasting accuracy. In order to simulate scenarios with varying weather forecast accuracy, normally-distributed random errors with

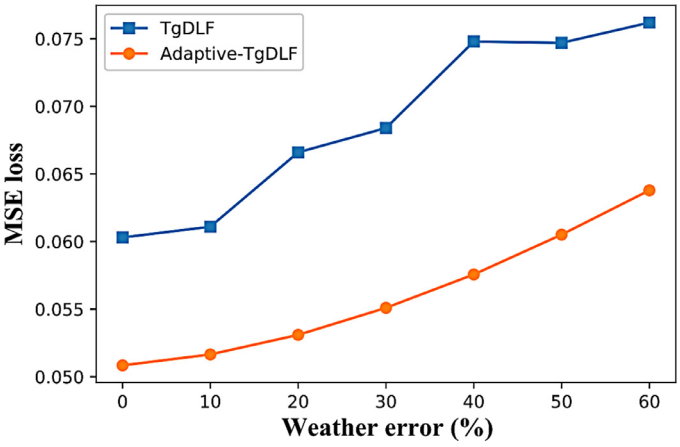

Fig. 9. The MSE loss of TgDLF and Adaptive-TgDLF with different scales of noise in the weather forecast data.

standard deviations of 10%, 20%, 30%, 40%, 50%, and 60%, respectively, are added to the weather data of the test set while testing.

The experimental results are displayed in Fig. 9. The normally-distributed error added to the weather forecast data appears on the x-axis, while the MSE loss appears on the y-axis. As the amount of noise in the weather forecast data grows, it is evident from Fig. 9 that both TgDLF and Adaptive-TgDLF's predictive capabilities decline as predicted, but Adaptive-TgDLF's rate of decline is slower, demonstrating that it is comparatively more robust to weather noise. Furthermore, even if the proportion of weather noise reaches 50%, the performance of Adaptive-TgDLF is still the same as that of TgDLF without weather noise, which further reflects the high accuracy and robustness of Adaptive-TgDLF. Detailed forecast results for different districts are given in Appendix E.

Fig. 10. The similarity between loads of different districts.

3.4. Transfer learning experiments

In transfer learning experiments, we choose the load data of a district as the target data and the load data of another district as the source data. The goal is to improve the performance of the model in the target data. In transfer learning, the source data are often sufficient, while the target data are not sufficient. Due to the difference in the distribution of source data and target data, it is difficult to directly combine the source data and the target data to train the deep-learning model. To match the transfer learning scenario [31–33], we only select 192 d of load data in the source district, and then select separately 128, 64, 32, and 16 d of load data in the target district for training. We also select 64 d of load data in the target district for testing. The source data and the target data are selected in order from 2008.01. Noise is also added to weather data to test the robustness of the model.

The maximum mean discrepancy (MMD) algorithm [59,60] can be used to measure the similarity between load data in different districts. Fig. 10 shows the MMD distance of the load data in 12 districts. Overall, the smaller is the value, the higher is the similarity of the load data of the two districts. In each group, the load data with high similarity are selected for the transfer learning experiment. We finally choose three sets of districts (MTG, FS), (MY, PG), and (CY, FT) (the first district in a set is the district of the source data, and the latter is the district of the target data), and transfer learning achieved good results on all three sets of load data. As an example, the set (MTG, FS) is discussed next.

Table 3
Prediction MSE (load ratio) of the initial model with and without transfer learning.

	AVG	Exp 0	Exp 1	Exp 2	Exp 3	Exp 4
No Transfer	0.190	0.100	0.078	0.167	0.331	0.275
Transfer	**0.008**	**0.007**	**0.008**	**0.008**	**0.007**	**0.008**

To ensure the objectivity of the experiment, we conduct five independent repeated experiments (the source data and the target data used in the five experiments are the same), and the parameters of the model are randomly initialized in each experiment. We can observe how the initial model performs on the target test set with and without transfer learning.

It is shown in Table 3 that the initial loss of the model is high when transfer learning is not used for initialization, and the loss caused by different initialization parameters is markedly different. After using transfer learning, the target data can be optimized from a very low loss, and it

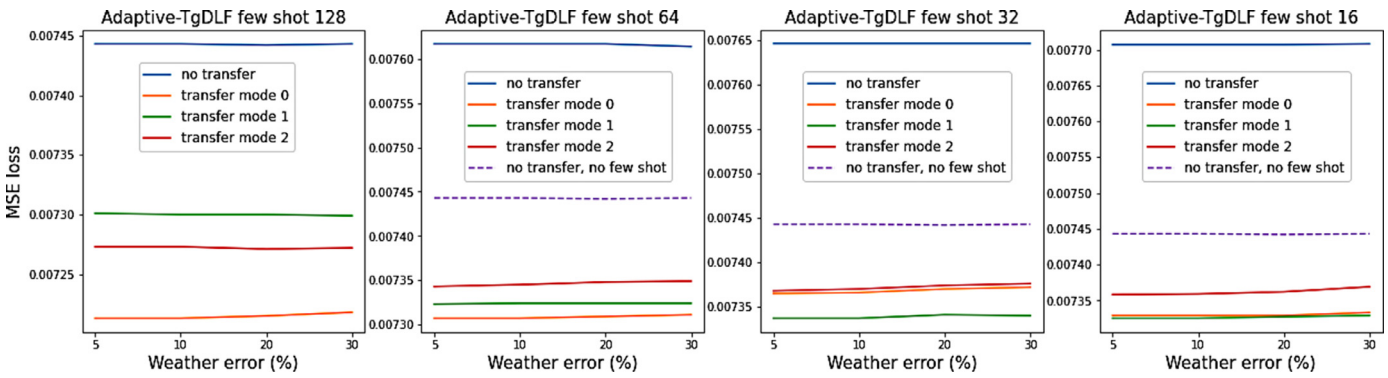

Fig. 11. Transfer learning results of the set (MTG, FS).

only needs half of the training epochs to converge compared to no transfer learning. Transfer learning can provide better initialization, which not only makes convergence faster, but also makes training more stable (the initial losses of the model on target data are similar). The loss curve is provided in Appendix F.

The final performances of the model under different transfer strategies (i.e., strategy 1 is updating all weights of the model while re-training; strategy 2 is only updating the weights of the last fully connected layer while re-training; and strategy 3 is only updating the weights of the encoder, decoder, and embedding layers while re-training), different scales of weather noise, and different samples of target load data are also illustrated in Fig. 11.

Transfer learning on the set (MTG, FS) outperforms no transfer learning, as shown in Fig. 11. The added weather noise is normally-distributed noise with a mean of zero, and the variance of the weather noise is represented by the abscissa in Fig. 11. As the variance of weather noise increases, the performance of Adaptive-TgDLF does not decrease obviously, which proves that Adaptive-TgDLF possesses strong robustness. As the target training data decrease, the performance of the model will decrease, but the effect of transfer learning is still significantly better than no transfer learning. The first transfer learning strategy is relatively the best among the three strategies for the set (MTG, FS). The transfer learning results of the set (MY, PG) and the set (CY, FT) are also provided in Appendix F.

3.5. Online learning experiments

In online learning experiments, we choose the early load data of a district as the historical data and the late load data of the district as the latest online data. The goal is to adapt the model to the latest data distribution of load and improve the forecasting performance of the model in the latest load data. To match the online learning scenario [34,35], we select the first 192 d of load data of a district in 2008 as the historical load data, and then select separately the first 192 d of load data of the district in 2009 and 2010 as the latest online load data. The load data of the remaining days in each of the three years in this district are used as the test set. We first train an offline model on the historical load data of 2008, and then test the model on the test set of 2008. Next, we continue to train the offline model on the online load data of 2009 and 2010, respectively, to obtain the online models adapted to the latest data, and then test the models on the test set of 2009 and 2010, respectively. To ensure the objectivity of the experiment, we tested the online learning method in three different districts (FT, PG, and FS).

We conduct 10 independent repeated experiments in each district, and the parameters of the model are randomly initialized in each experiment. We take the average value of 10 experiments as the forecasting result of the model on the test set, and the online learning results of different districts are shown in Fig. 12. It can be seen from Fig. 12 that the performance of offline models in the three districts has been greatly improved after online learning with the load data of 2009, and further

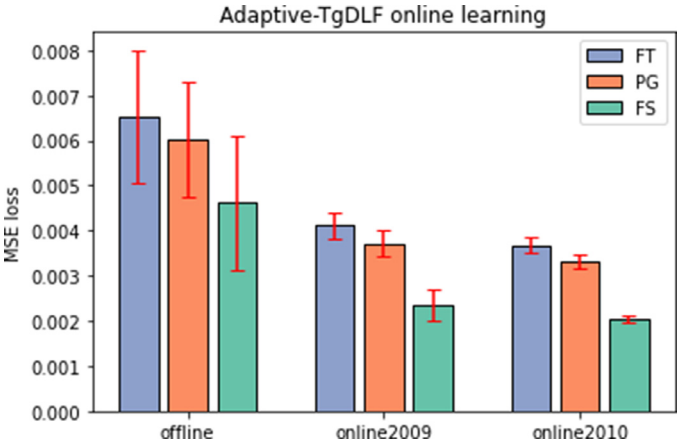

Fig. 12. Online learning results of different districts.

improved after online learning with load data of 2010, which proves the effectiveness of online learning.

4. Conclusion

In this study, we propose Adaptive-TgDLF, a more accurate and efficient model for short-term electrical load forecasting. We utilize historical load data, weather factors, and calendar factors to predict local fluctuations, while obtaining dimensionless trends through expert experience or historical data. Adaptive-TgDLF achieves superior performance, leading to the following contributions: (1) it reduces the average MSE from 0.061 to 0.051, representing a 16% increase in accuracy over the previous version TgDLF. Additionally, Adaptive-TgDLF maintains good accuracy even with a 50% weather noise, demonstrating its robustness; (2) Adaptive-TgDLF successfully integrates adaptive learning methods to improve the generalization ability of the load forecasting model in location and time. It uses transfer learning to address limited data, as well as online learning to address load distribution changes over time. Adaptive learning allows the load data of different spaces or times to assist each other in training a more efficient model; (3) Adaptive-TgDLF reduces average training time from 769 s to 344 s compared to TgDLF, saving over 50% of training time; and (4) the model Transformer in Adaptive-TgDLF offers some level of interpretability, providing better theory guidance.

To our knowledge, Adaptive-TgDLF is the first application that combines the model Transformer with human knowledge and mines the interpretability of Transformer for short-term electrical load forecasting. It is also the first model to use adaptive learning to address issues of insufficient historical load data and load distribution changes. This work is innovative and may contribute meaningfully to future research efforts in this field.

5. Generalization for other forecasting tasks and future work

Adaptive-TgDLF is not limited to load forecasting but can also be applied to other forecasting tasks. The model Transformer is effective in capturing long-term dependencies in time-series data, which is applicable to all time-series forecasting tasks. Additionally, time-series data typically exhibits dimensionless trends and local fluctuations, making series decomposition, as employed in Adaptive-TgDLF, useful for all types of time-series data. The dimensionless trend can be obtained using the filtered average trend as mentioned in the paper, or extracted using more complex expert knowledge tailored to the specific task. Transfer learning and online learning have also proven to be effective in multiple tasks, and thus, adaptive learning is likewise beneficial for various time-series forecasting tasks. From our perspective, the only modification required for applying Adaptive-TgDLF to different tasks may be the preprocessing of the series data: for series with strong periodicity like the electrical load, such as electricity transformers temperature or traffic data, the ratio conversion used in the paper can be directly adopted; but for other series with weaker periodicity, such as weather or exchange rate data, the ratio conversion may not achieve data stationarity, and therefore standardization or other preprocessing methods may be necessary.

We consider the following as future work for Adaptive-TgDLF: (1) the current dimensionless trend is a simple filtered weekly average trend, and it is possible to choose a more complicated and accurate dimensionless trend; (2) the current loss function for model training is the ordinary MSE loss function, which only considers the numerical difference between the predicted value and the actual value. It is possible to further add human knowledge to the loss function in the form of regularization terms, and it is also possible to directly embed domain knowledge in the model structure; (3) the functions of interpretability deserve further exploration, which may guide model training to obtain better performance. For instance, it is possible to judge the convergence of the model based on the shape of the attention matrix, and weighting the information of different time periods is also a possible method to guide model training; and (4) the current attention matrix of Transformer is utilized to learn the interaction between all features at different time-steps. It is possible to construct a 3D attention matrix to learn the interaction between different features at different time-steps. If one of these proposals is proven to work, Adaptive-TgDLF will achieve better performance.

CODE availability

The code of Adaptive-TgDLF is made available for download through the following link: https://github.com/daxin007/Adaptive-TgDLF

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work is funded by the National Natural Science Foundation of China (Grant No. 62106116), and the SUSTech - Qingdao New Energy Technology Research Institute.

APPENDIX A. FUNCTIONS AND ADVANTAGES OF RATIO CONVERSION

Since the load series is periodic and the load values at time t is similar to that at t + 1 (t + 1 represents the same time-step of the next day), the value of the load ratio is around 1. Therefore, this conversion offers three main functions: (1) it can convert non-stationary load series into stationary load ratio series; (2) it can compress the value range of features to a relatively small interval, which can speed-up convergence of the models; and (3) it can move the load data in different districts to a similar interval, which is similar to standardization, and can enhance the generalization ability of the models. Furthermore, the ratio conversion may have two advantages over standardization: (1) the mean and standard deviation of the training set are required and directly utilized for standardization on the test set. When the distributions of the training set and the test set are different, the mean and standard deviation will become inaccurate, and the converted training set and test set are not in the same value range. However, the ratio conversion does not have this problem; (2) standardization is vulnerable to extreme values/outliers, while ratio conversion is not much affected by them. Here are examples of these two advantages:

Example 1. Suppose that the training set x_{train} is a continuous series which is generated by the sine function $f_{train}(x) = \sin(wx) + Noise$, where *Noise* satisfies a normal distribution N(0, 1), and the test set x_{test} is a continuous series which is generated by the sine function $f_{test}(x) = \sin(wx) + Noise + 5$, where *Noise* also satisfies a normal distribution N(0, 1). The standardization of the training set and the test set can be defined as follows:

$$x'_{train} = \frac{x_{train} - \mu_1}{\sigma_1}$$
$$x'_{test} = \frac{x_{test} - \mu_1}{\sigma_1} \quad (5)$$

where μ_1 and σ_1 are the calculated mean and standard deviation of the training set x_{train}, respectively; x'_{train} and x'_{test} represent the standardized training set and test set, respectively; and x'_{train} satisfies the normal distribution N(0, 1), but x'_{test} does not, and thus they are not in the same value range.

The ratio conversion is defined in Eq. (1). After the ratio conversion of the training set x_{train} and the test set x_{test}, the converted values are around 1, and thus the value range of the ratio-converted training set and test set are the same.

Example 2. Suppose that the training set x_{train} is a continuous series with 10,000 samples which satisfies a normal distribution N(0, 1). If there is an outlier with a value of 20,000 in the training set, the calculated mean value of the training set will be about 2, which is obviously inaccurate. The subsequent standardization will also become inaccurate with this inaccurate mean value. However, the ratio conversion can avoid this problem: after the ratio conversion, only the ratio value with this outlier is abnormal, while the other ratio values are normal. In ratio conversion, outliers only affect local values; while in standardization conversion, outliers affect all values, so standardization is more vulnerable to extreme values/outliers.

APPENDIX B. THE DETAILED INPUT AND OUTPUT OF TRANSFORMER

The Transformer's input has three dimensions: (1) batch size, which is the number of samples used in each training iteration, and the batch size used in training is 512; (2) time-step, which refers to how long the historical load is used for training, and the time-step used is 96 h; and (3) the feature dimension, which contains a historical data feature together with four weather features and four date features, for a total of nine features. Transformer contains an encoder block and a decoder block, and the dimension of the hidden layer is 12. For the attention module, each attention is divided into two heads, and these two heads may learn different information while training. The attention module in Transformer is utilized to learn the interaction between all features at different time-steps. The Transformer's output also has three dimensions: (1) batch size, which is 512 and consistent with the input's batch

Table C1
Prediction RMSE of different models.

	ARMA	ARIMA	LSTM	DT	EnLSTM	TgDLF	Transformer	Adaptive-TgDLF
PG	0.510	0.336	0.319	0.348	0.308	0.278	0.267	**0.247**
SJS	0.640	0.339	0.339	0.340	0.346	0.326	0.320	**0.311**
CY	0.266	0.249	0.227	0.256	0.215	0.178	0.165	**0.152**
YZ	0.400	0.348	0.302	0.354	0.298	0.283	0.261	**0.236**
MTG	1.515	0.569	0.332	0.351	0.346	0.327	0.323	**0.310**
FT	0.311	0.265	0.240	0.253	0.229	0.173	0.172	**0.167**
MY	0.434	0.288	0.269	0.280	0.255	0.222	0.224	**0.210**
FS	0.602	0.319	0.305	0.310	0.302	0.274	0.281	**0.266**
CP	0.379	0.263	0.243	0.236	0.229	0.199	0.194	**0.186**
SY	0.417	0.302	0.280	0.317	0.265	0.234	0.220	**0.203**
HD	0.285	0.279	0.252	0.284	0.231	0.204	0.196	**0.176**
DX	0.389	0.232	0.243	0.248	0.226	0.187	0.183	**0.167**
AVG	0.605	0.327	0.279	0.298	0.271	0.241	0.234	**0.219**

Table C2
Prediction MAPE of different models.

	ARMA	ARIMA	LSTM	DT	EnLSTM	TgDLF	Transformer	Adaptive-TgDLF
PG	–	–	0.044	0.047	0.042	0.038	0.036	**0.033**
SJS	–	–	0.065	0.061	0.066	0.057	0.056	**0.055**
CY	–	–	0.050	0.048	0.042	0.035	0.032	**0.030**
YZ	–	–	0.051	0.056	0.050	0.048	0.043	**0.039**
MTG	–	–	0.062	0.064	0.064	0.060	0.060	**0.057**
FT	–	–	0.049	0.047	0.044	0.034	0.033	**0.031**
MY	–	–	0.043	0.042	0.039	0.035	0.033	**0.031**
FS	–	–	0.040	0.040	0.040	0.036	0.036	**0.034**
CP	–	–	0.050	0.045	0.045	0.038	0.037	**0.035**
SY	–	–	0.047	0.050	0.043	0.038	0.035	**0.032**
HD	–	–	0.054	0.050	0.044	0.037	0.035	**0.033**
DX	–	–	0.053	0.049	0.045	0.037	0.035	**0.032**
AVG	–	–	0.051	0.050	0.047	0.041	0.039	**0.037**

size; (2) time-step, which refers to how long to forecast the load in the future, and the time-step used is 24 h; and (3) the feature dimension, in which the feature dimension is 1 since the model only forecasts the load here.

APPENDIX C. PREDICTION RMSE AND MAPE OF DIFFERENT MODELS

The prediction RMSE of different models is shown in Table C.1, and the prediction MAPE of different models is shown in Table C.2. It can be seen from the tables that Adaptive-TgDLF also has an obvious advantage in each district under these two metrics.

APPENDIX D. EVOLVING PROCESS OF THE ATTENTION MATRIX

The evolving of the attention matrices of a sample in the training process is shown in Fig. D.1. From left to right, the sample is in the 0th epoch (untrained initial state), the 10th epoch, the 50th epoch, and the 100th epoch. It can be found that the attention matrix of the sample gradually evolves from the initial checkerboard-shaped to band-shaped. After 100 epochs (the total training epochs are 300), the attention matrices of this sample will always be band-shaped. In fact, we find that the evolving processes of attention matrices of most samples are from checkerboard-shaped (or bright spot-shaped) to band-shaped. There are also a few samples whose attention matrix remain checkerboard-shaped (or bright spot-shaped), which may prove that they are not best fitted by the model Transformer (in general, the deep-learning model cannot fit all samples in the training set the best, unless it is over-fitted), or the data distribution of the samples is inconsistent with that of most samples.

APPENDIX E. DETAILED FORECAST EFFECTS

Detailed forecast effects of Adaptive-TgDLF with different scales of noise in the weather forecast data for different districts are listed in Table E.1. The header of the table represents the scale of normally-distributed error added to the weather forecast data.

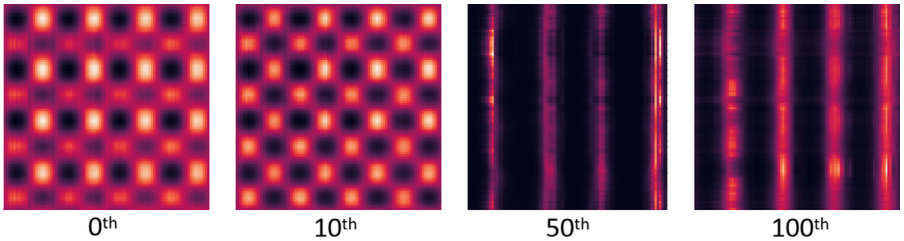

Fig. D1. The evolving of the attention matrices of a sample.

Table E1
The MSE loss of Adaptive-TgDLF with different scales of noise in the weather forecast data.

	0	10%	20%	30%	40%	50%	60%
PG	0.061	0.062	0.064	0.066	0.070	0.074	0.078
SJS	0.097	0.097	0.098	0.099	0.100	0.101	0.102
CY	0.023	0.023	0.024	0.026	0.028	0.030	0.033
YZ	0.056	0.059	0.062	0.066	0.070	0.073	0.077
MTG	0.096	0.097	0.098	0.099	0.100	0.101	0.102
FT	0.028	0.028	0.030	0.033	0.037	0.040	0.044
MY	0.044	0.046	0.047	0.049	0.052	0.055	0.057
FS	0.071	0.072	0.073	0.074	0.076	0.077	0.078
CP	0.035	0.035	0.037	0.040	0.042	0.045	0.049
SY	0.041	0.042	0.043	0.045	0.049	0.053	0.058
HD	0.031	0.031	0.031	0.032	0.033	0.033	0.034
DX	0.028	0.028	0.029	0.032	0.036	0.043	0.052
AVG	0.051	0.052	0.053	0.055	0.058	0.061	0.064

APPENDIX F. ADDITIONAL RESULTS OF TRANSFER LEARNING

The loss curve of the set (MTG, FS) is shown in Fig. F.1. It can be seen that the target data can be optimized from a very low loss with transfer learning, and it only needs half of the training epochs to converge compared to no transfer learning. The transfer learning results of the set (MY, PG) and the set (CY, FT) are shown in Fig. F.2 and Fig. F.3, respectively. The performances of transfer learning on these two sets are also significantly better than those of no transfer learning. The third transfer learning strategy (i.e., update the weights of the encoder, decoder, and embedding layers of the model while re-training) is relatively the best among the three strategies for the set (MY, PG), and the second transfer learning strategy (i.e., update the weights of the last fully connected layer of the model while re-training) is relatively the best for the set (CY, FT). Overall, the effects of the three strategies are not substantially different.

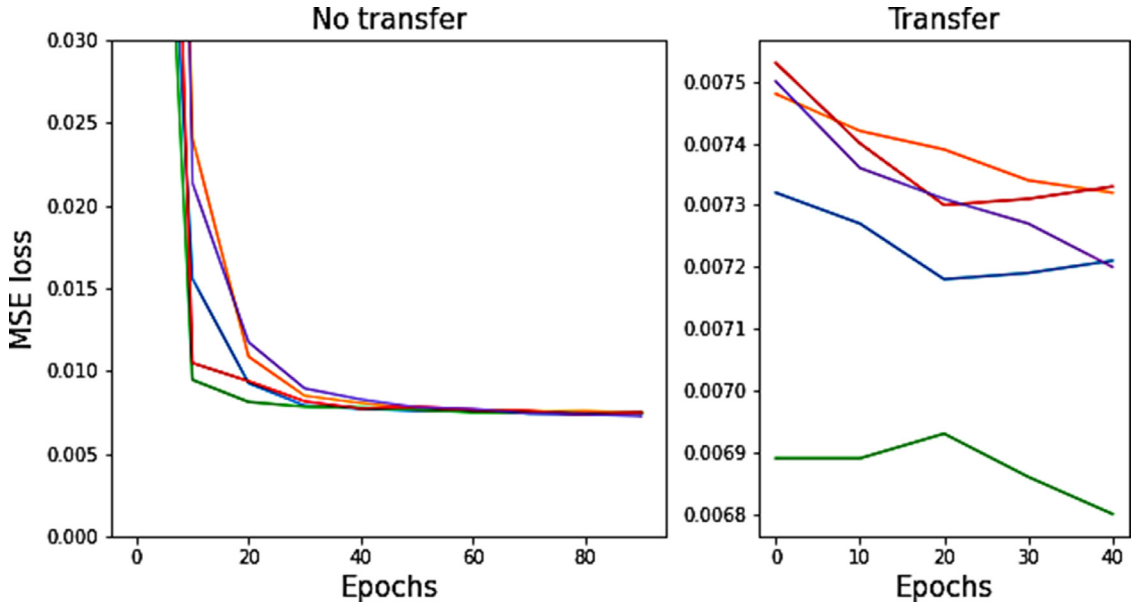

Fig. F1. Loss curve of the set (MTG, FS).

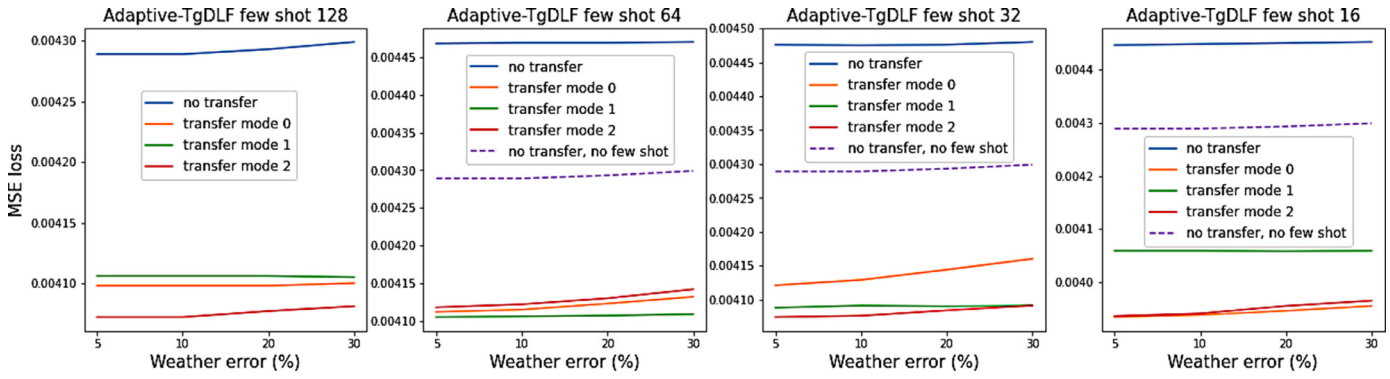

Fig. F2. Transfer learning results of the set (MY, PG).

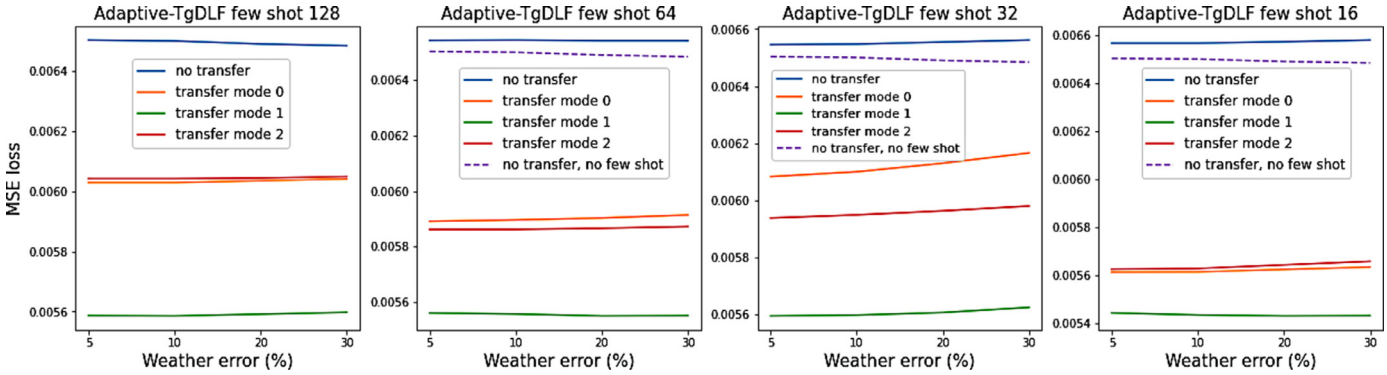

Fig. F3. Transfer learning results of the set (CY, FT).

References

[1] Fallah SN, Ganjkhani M, Shamshirband S, et al. Computational intelligence on short--term load forecasting: a methodological overview[J]. Energies 2019;12(3):393.
[2] Rahman S, Bhatnagar R. An expert system based algorithm for short term load forecast[J]. IEEE Trans Power Syst 1988;3(2):392–9.
[3] Papalexopoulos AD, Hesterberg TC. A regression-based approach to short-term system load forecasting[J]. IEEE Trans Power Syst 1990;5(4):1535–47.
[4] Hassan S, Khosravi A, Jaafar J, et al. A systematic design of interval type-2 fuzzy logic system using extreme learning machine for electricity load demand forecasting[J]. Int J Electrical Power Energy Syst 2016;82:1–10.
[5] Ali D, Yohanna M, Puwu MI, et al. Long-term load forecast modelling using a fuzzy logic approach[J]. Pacific Sci Rev A: Natural Sci Eng 2016;18(2):123–7.
[6] Lindberg O, Lingfors D, Arnqvist J, et al. Day-ahead probabilistic forecasting at a co-located wind and solar power park in Sweden: trading and forecast verification[J]. Adv Appl Energy 2023:100120.
[7] DebRoy T, Mukherjee T, Wei HL, et al. Metallurgy, mechanistic models and machine learning in metal printing[J]. Nature Rev Mater 2021;6(1):48–68.
[8] Goodell JW, Kumar S, Lim WM, et al. Artificial intelligence and machine learning in finance: identifying foundations, themes, and research clusters from bibliometric analysis[J]. J Behav Exp Finance 2021;32:100577.
[9] Wang C, Peng K. AI Experience Predicts Identification with Humankind[J]. Behav Sci 2023;13(2):89.
[10] Park DC, El-Sharkawi MA, Marks RJ, et al. Electric load forecasting using an artificial neural network[J]. IEEE Trans Power Syst 1991;6(2):442–9.
[11] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Comput 1997;9(8):1735–80.
[12] Bedi J, Toshniwal D. Deep learning framework to forecast electricity demand[J]. Appl Energy 2019;238:1312–26.
[13] Shi H, Xu M, Li R. Deep learning for household load forecasting—A novel pooling deep RNN[J]. IEEE Trans Smart Grid 2017;9(5):5271–80.
[14] Ouyang T, He Y, Li H, et al. Modeling and forecasting short-term power load with copula model and deep belief network[J]. IEEE Trans Emerging Topics in Comput Intell 2019;3(2):127–36.
[15] Dai Y, Zhao P. A hybrid load forecasting model based on support vector machine with intelligent methods for feature selection and parameter optimization[J]. Appl Energy 2020;279:115332.
[16] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Adv Neural Inf Process Syst 2017:30.
[17] Jurasovic M, Franklin E, Negnevitsky M, et al. Day ahead load forecasting for the modern distribution network-a Tasmanian case study[C]. In: 2018 Australasian Universities Power Engineering Conference (AUPEC). IEEE; 2018. p. 1–6.
[18] Kasongo SM, Sun Y. A deep learning method with filter based feature engineering for wireless intrusion detection system[J]. IEEE Access 2019;7:38597–607.
[19] Wei W, Hu X, Liu H, et al. Towards Integration of Domain Knowledge-Guided Feature Engineering and Deep Feature Learning in Surface Electromyography-Based Hand Movement Recognition[J]. Comput Intell Neurosci 2021.
[20] Wang N, Zhang D, Chang H, et al. Deep learning of subsurface flow via theory-guided neural network[J]. J Hydrol (Amst) 2020;584:124700.
[21] He T, Zhang D. Deep learning of dynamic subsurface flow via theory-guided generative adversarial network[J]. J Hydrol (Amst) 2021;601:126626.
[22] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J]. J Comput Phys 2019;378:686–707.
[23] Karpatne A, Atluri G, Faghmous JH, et al. Theory-guided data science: a new paradigm for scientific discovery from data[J]. IEEE Trans Knowl Data Eng 2017;29(10):2318–31.
[24] Daw A, Karpatne A, Watkins WD, et al. Physics-guided neural networks (pgnn): an application in lake temperature modeling[M]. In: Knowledge-Guided machine learning. Chapman and Hall/CRC; 2017. p. 353–72.
[25] He T, Wang N, Zhang D. Theory-guided full convolutional neural network: an efficient surrogate model for inverse problems in subsurface contaminant transport[J]. Adv Water Resour 2021;157:104051.
[26] Li J, Zhang D, Wang N, et al. Deep Learning of Two-Phase Flow in Porous Media via Theory-Guided Neural Networks[J]. SPE Journal 2022;27(02):1176–94.
[27] Chen Y, Huang D, Zhang D, et al. Theory-guided hard constraint projection (HCP): a knowledge-based data-driven scientific machine learning method[J]. J Comput Phys 2021;445:110624.
[28] Chen Y, Zhang D. Theory-guided deep-learning for electrical load forecasting (TgDLF) via ensemble long short-term memory[J]. Adv Appl Energy 2021;1:100004.
[29] Chen Y, Zhang D. Well log generation via ensemble long short-term memory (EnLSTM) network[J]. Geophys Res Lett 2020;47(23):e2020GL087685.
[30] Chen Y, Chang H, Meng J, et al. Ensemble Neural Networks (ENN): a gradient-free stochastic method. Neural Networks, 2019;110:170–85.
[31] Pan SJ, Yang Q. A survey on transfer learning[J]. IEEE Trans Knowl Data Eng 2009;22(10):1345–59.
[32] Luo X, Zhang D, Zhu X. Combining transfer learning and constrained long short--term memory for power generation forecasting of newly-constructed photovoltaic plants[J]. Renew Energy 2022;185:1062–77.
[33] Pinto G, Wang Z, Roy A, et al. Transfer learning for smart buildings: a critical review of algorithms, applications, and future perspectives[J]. Adv Appl Energy 2022:100084.
[34] Hoi SCH, Sahoo D, Lu J, et al. Online learning: a comprehensive survey[J]. Neurocomputing 2021;459:249–89.
[35] Fekri MN, Patel H, Grolinger K, et al. Deep learning for load forecasting with smart meter data: online Adaptive Recurrent Neural Network[J]. Appl Energy 2021;282:116177.
[36] Asare-Bediako B, Kling WL, Ribeiro PF. Day-ahead residential load forecasting with artificial neural networks using smart meter data[C]. In: 2013 IEEE Grenoble Conference. IEEE; 2013. p. 1–6.
[37] Lusis P, Khalilpour KR, Andrew L, et al. Short-term residential load forecasting: impact of calendar effects and forecast granularity[J]. Appl Energy 2017;205:654–669.
[38] Taylor JW, Buizza R. Neural network load forecasting with weather ensemble predictions[J]. IEEE Trans Power Syst 2002;17(3):626–32.
[39] Lin T, Wang Y, Liu X, et al. A survey of transformers[J]. arXiv preprint 2021. arXiv:2106.04554.
[40] Chen Q, Zhao H, Li W, et al. Behavior sequence transformer for e-commerce recommendation in alibaba[C]. In: Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data; 2019. p. 1–4.
[41] Charton F. Linear algebra with transformers[J]. arXiv preprint 2021. arXiv:2112.01898.
[42] Devlin J, Chang MW, Lee K, et al. Bert: pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint 2018. arXiv:1810.04805.
[43] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Adv Neural Inf Process Syst 2020;33:1877–901.
[44] Ramesh A, Pavlov M, Goh G, et al. Zero-shot text-to-image generation[C]. In: International Conference on Machine Learning. PMLR; 2021. p. 8821–31.
[45] Chen M, Tworek J, Jun H, et al. Evaluating large language models trained on code[J]. arXiv preprint 2021. arXiv:2107.03374.
[46] Wen Q, Zhou T, Zhang C, et al. Transformers in Time Series: a Survey[J]. arXiv preprint 2022. arXiv:2202.07125.
[47] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 770–8.
[48] Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning[J]. Neurocomputing 2021;452:48–62.
[49] Chen Z, Xiao F, Guo F, et al. Interpretable machine learning for building energy management: a state-of-the-art review[J]. Adv Appl Energy 2023:100123.
[50] Chan W, Jaitly N, Le QV, et al. Listen, attend and spell[J]. arXiv preprint 2015. arXiv:1508.01211.
[51] Baan J, ter Hoeve M, van der Wees M, et al. Understanding multi-head attention in abstractive summarization[J]. arXiv preprint 2019. arXiv:1911.03898.
[52] Wang X, Xiong Y, Qian X, et al. Lightseq2: accelerated training for transformer-based models on gpus[J]. arXiv preprint 2021. arXiv:2110.05722.

[53] Wong T, Yeh PY. Reliable accuracy estimates from k-fold cross validation[J]. IEEE Trans Knowl Data Eng 2019;32(8):1586–94.
[54] Valipour M, Banihabib ME, Behbahani SMR. Comparison of the ARMA, ARIMA, and the autoregressive artificial neural network models in forecasting the monthly inflow of Dez dam reservoir[J]. J Hydrol (Amst) 2013;476:433–41.
[55] Chujai P, Kerdprasop N, Kerdprasop K. Time series analysis of household electric consumption with ARIMA and ARMA models[C]. In: Proceedings of the International Multiconference of Engineers and Computer Scientists, 1. Hong Kong: IAENG; 2013. p. 295–300.
[56] Pong PWT, Annaswamy AM, Kroposki B, et al. Cyber-enabled grids: shaping future energy systems[J]. Adv Appl Energy 2021;1:100003.
[57] Feng C, Wang Y, Chen Q, et al. Smart grid encounters edge computing: opportunities and applications[J]. Adv Appl Energy 2021;1:100006.
[58] Luo J, Hong T, Fang SC. Benchmarking robustness of load forecasting models under data integrity attacks[J]. Int J Forecast 2018;34(1):89–104.
[59] Smola AJ, Gretton A, Borgwardt K. Maximum mean discrepancy[C]. In: 13th International Conference. Hong Kong, China: ICONIP; 2006. October 3-6, 2006: Proceedings. 2006.
[60] Long M, Zhu H, Wang J, et al. Deep transfer learning with joint adaptation networks[C]. In: International Conference on Machine Learning. PMLR; 2017. p. 2208–17.