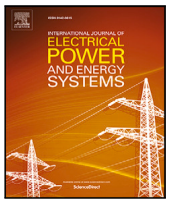
Electrical Power and Energy Systems 172 (2025) 111195

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

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



Short-term load forecasting method for integrated energy systems based on graph neural network and multi-task balance

Chen Wang [a,c], Ying Wang [a,*], Enrico Zio [b,c], Kaifeng Zhang [a]

[a] Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, China
[b] Mines Paris-PSL University, CRC, Sophia Antipolis, France
[c] Department of Energy, Politecnico di Milano, Milano, Italy

ARTICLE INFO

Keywords:
Load forecasting
Integrated energy system
Multi-energy loads
Graph neural network
Multi-task balance

ABSTRACT

Electricity load, heat load and cold load in integrated energy systems are related to each other. Inspired by Graph Neural Networks (GNN), which can capture the topological structure of graph data, this paper proposes a novel multi-energy forecasting framework with multi-level task-sharing matrices (MTMs), which connects multiple tasks to capture the coupling characteristics between multiple loads. The proposed framework overwhelms most traditional multi-task approaches because traditional models usually have shared parameters in only one stage, which reserves representation space for temporal features of individual loads but limits their ability to capture multi-task correlations across multiple stages. Specifically, the proposed framework can capture coupling features across various stages of the entire model by MTMs, significantly enhancing the connections among multiple tasks while preserving adequate representation space for temporal features. The combination of graph attention and graph convolution can further expand the representational space of coupling features, so that multi-head attention can focus on capturing the temporal characteristics of individual loads. In addition, a gradient-based multi-task balance method is proposed to adapt to the above framework, which can balance different tasks by normalizing the weights by MTMs. Case studies demonstrate that the proposed framework has superior forecasting performance for multi-energy loads.

1. Introduction

Integrated energy systems can make use of the complementary advantages of different energy sources to achieve the efficient utilization of multiple energies and stable supply for energy consumption [1]. In integrated energy systems, the relation between multiple energy carriers can be strong [2]. Therefore, for multi-energy load forecasting, it is necessary to consider the interaction and coupling information between the multiple energy sources [3].

For traditional single energy systems, energy forecasting often only focuses on one type of energy [4]. Regarding the power load forecasting problem, Li et al. [5] proposed a mid-term forecasting method based on manifold learning and Long Short Term Memory (LSTM) networks to reduce the calculation burden. Regarding the forecasting of wind power with high volatility, Khodayar et al. [6] proposed a deep architecture with Stacked Auto Encoder (SAE) and Stacked Denoising Auto Encoder (SDAE). This deep network architecture is used to capture intense temporal fluctuations. Considering the information of neighboring houses, Lin et al. [7] used Graph Neural Network (GNN) to build a short-term residential load forecasting model. For the same multivariate residential load forecasting problem, Zhao et al. [8] used the Spatio-Temporal Graph Attention Transformer (STGAT) model to capture the correlation patterns among multiple loads. Jiang et al. [9] proposed a multi-task forecasting method with two levels of information extraction for multiple individual household based on Convolutional Neural Network (CNN) and LSTM. Considering the correlation between the reactive and active loads, Qin et al. [10] proposed a multi-task forecasting method based on LSTM. The multi-task frameworks in [9,10] are both hard parameter sharing. Therefore, the multi-task model, which can capture the correlation between multiple factors, can be an effective method to solve the problem of multi-energy load forecasting.

Recently, multi-energy load forecasting techniques have been studied. Basically, most of them use the hard parameter sharing method to capture the coupling relationship among multi-energy loads. Considering the multi-energy coupling relation, Niu et al. [13] proposed a multi-task model with hard weight sharing based on one-dimensional CNN and Bidirectional Gated Recurrent Unit (BiGRU). Guo et al. [12] analyzed the correlation of multi-energy loads in different seasons

* Corresponding author.
E-mail address: wyseu@seu.edu.cn (Y. Wang).

https://doi.org/10.1016/j.ijepes.2025.111195
Received 11 March 2025; Received in revised form 6 July 2025; Accepted 24 September 2025
Available online 30 September 2025

Table 1
Typical references for load forecasting.

Paper	Multi-task	Load	Model	Coupling method	Task balance
[4]	×	Electrical	LSTM+Attention	–	–
[5]	×	Electrical	LSTM	–	–
[7]	×	Electrical	TCN+GNN	Top Sharing Adjacency Matrix	–
[9]	✓	Multi-household Electrical	CNN+LSTM	Middle Sharing	×
[10]	✓	Reactive and Active Electrical	LSTM	Bottom Sharing	×
[11]	✓	Multi-energy	CNN+LSTM	Middle Sharing	✓
[12]	✓	Multi-energy	BiLSTM	Middle Sharing	×
[13]	✓	Multi-energy	CNN+BiGRU	Bottom Sharing	✓
[14]	✓	Multi-energy	CNN+BiLSTM	Bottom Sharing	×
[15]	✓	Multi-energy	Multi-model Fusion	Separate models	✓
[16]	✓	Multi-energy	STA+GTCN	Bottom Sharing	×
[17]	✓	Multi-energy	LSTM+TCN	Middle Sharing	×
[18]	✓	Multi-energy	DBN	Middle Sharing	×
[19]	✓	Multi-energy	CNN+GRU+GBDT	Bottom Sharing	×
[20]	✓	Multi-energy	CNN+GRU	Top Sharing	✓
[21]	✓	Multi-energy	Transformer	Separate encoder	×
[22]	✓	Multi-energy	Bayesian Transformer	Separate encoder	✓
[23]	✓	Multi-energy	TCN+GNN	Top Sharing Adjacency Matrix	×
Proposed	✓	Multi-energy	Attention+GNN	Multi-level Task-sharing Matrices	✓

by Maximum Information Coefficient (MIC) and proposed a multi-task model with hard weight sharing based on Bi-directional LSTM (BiLSTM). The decomposition-reconstruction framework performs well in the problem of multi-energy load forecasting. Li et al. [14] used the feature separation-fusion method to differentially process distinct features and proposed a short-term load forecasting method based on CNN and LSTM. After that, Lin et al. [24] used the Two-Layer Joint Modal Decomposition (TLJMD) method to decompose the loads into several Intrinsic Mode Functions (IMFs) and adopted the Dynamic Optimal Ensemble (DOE) learning method to provide the forecasting results. Differently, Hu et al. [25] used Empirical Mode Decomposition (EMD) for data filtering and reconstruction as feature engineering. Then, a forecasting model based on BiLSTM and Transformer is constructed. For the ultra-short-term forecasting problem, Qu et al. [26] combined the decomposition-reconstruction architecture with progressive layered extraction multi-task learning to achieve good results. Considering the attention to features in multiple dimensions, Song et al. [16] proposed a multi-task forecasting method based on the Spatio-Temporal Attention (STA) and Gated Temporal Convolutional Networks (GTCN). To handle the non-stationary sequence of multi-energy loads, Shi et al. [15] proposed a multi-model fusion forecasting method based on complex machine learning methods and heuristic algorithms. Considering the privacy reasons, Zhang et al. [11] proposed a privacy-preserving multi-energy load forecasting model based on federated learning. Taking into account the component of multi-energy load in the total energy demand, Wang et al. [17] used LSTM and Time Convolutional Networks (TCN) for temporal modeling. Considering the temporal dynamic features of multi-energy loads, Wang et al. [27] proposed an encoder–decoder model based on LSTM and Gradient Boosting Decision Tree (GBDT), which could effectively reflect the dynamic characteristics of historical loads. Considering the generation features of multi-energy prosumers, Zhou et al. [18] classified these prosumers into various aggregations, and proposed a multi-energy net load forecasting method based on deep belief network. Considering the multi-task modeling approach, Zhuang et al. [23] proposed a multi-energy load forecasting model based on TCN and GNN, which captures the correlation between multi-energy loads by one adjacency matrix and do not consider the balance between multiple tasks. Considering the high-dimensional temporal and spatial features, Xuan et al. [19] proposed a multi-task ensemble approach based on CNN, Gated Recurrent Unit (GRU) and GBDT, which could effectively extract features, model time series dynamically and achieve the sharing of prediction at different levels. Above methods may not fall under the typical multi-task learning networks, but they still consist of the bottom shared modules and specific task modules. Considering the issue of insufficient sample size, Li et al. [20] proposed a transfer learning-based prediction framework, which requires multiple time-consuming steps including model training, transfer learning and parameter fine tuning. In our previous research [21,22], the one-encoder multiple-decoder architecture was designed, attempting to decouple multiple tasks and separate the shared feature extraction process from each task. The attempt yielded good results but required much training time. The typical references for load forecasting are shown in Table 1.

To sum up, for enough representation space for temporal features of individual loads, multi-task models usually utilize only one stage of the model, such as the bottom layer [19], middle layer [18], top layer [20] or separate encoder [22], to capture the coupling features of multi-energy loads and use the rest parts of the model to capture the temporal features of individual loads.

However, there remains a research gap that traditional multi-task models usually have shared parameters in only one stage, which limits their ability to capture multi-task correlations across multiple stages. If the task-shared layers are too numerous or too large, the representation space for the temporal features of individual loads would become insufficient, thereby hindering their forecasting performance. If the coupling relationship can be established more efficiently across various stages of the entire model, it would enable a deeper capture of the coupling features across multiple dimensions, thereby significantly enhancing the connections among multiple load forecasting tasks while preserving adequate representation space for temporal features of individual loads.

To address this issue, a multi-energy load joint forecasting framework based on Multi-level Task-sharing Matrices (MTMs) is proposed. The proposed framework firstly introduces MTMs, equipping GNN with multi-level coupling components to capture deep relation features between different types of loads at different levels of abstraction. The combination of graph attention and graph convolution can further expand the representational space of coupling features, so that Multi-Head Attention (MHA) can focus on capturing the temporal characteristics of individual loads. In addition, to ensure that all subtasks benefit from the joint forecasting, a gradient-based balance method for multi-task weights is proposed to adapt to the above framework. The MTMs can capture the coupling information of multi-energy loads, so the gradient of the task weights can be normalized by the average of MTMs to balance the training of tasks in real-time, thereby achieving better performance and faster speed of joint forecasting.

As stated above, the contributions of this paper can be summarized as:

(1) A novel joint forecasting framework inspired by GNN is proposed for multi-energy load forecasting. This framework firstly introduces the multi-level task-sharing matrices, which equips the typical GNN with multi-level coupling components to capture

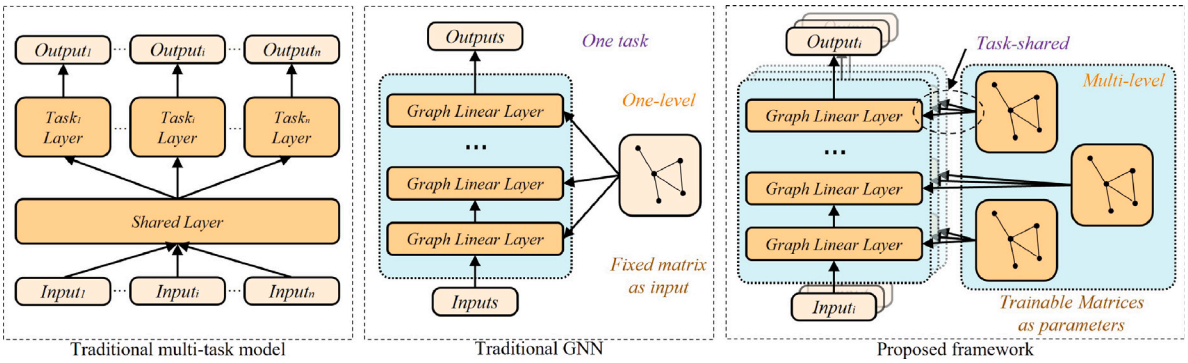


Fig. 1. Differences between the traditional multi-task model, the traditional GNN and the proposed framework.

deep coupling features between different loads at different levels of abstraction and maintains sufficient representation space for the temporal characteristics of loads.

(2) A gradient-based balance method for multi-task weights is proposed to adapt to the above framework. The gradients of task weights can be normalized by the average of task-sharing matrices to balance the training speed of multiple tasks in real-time and ensure that all subtasks benefit from the joint forecasting.

(3) Based on real-world data, the case study shows that the proposed framework has superior forecasting performance for multi-energy loads in terms of RMSE, MAPE and R^2 metrics.

The remainder of this paper is organized as follows. In Section 2, the novelty of the proposed framework and the motivation for improvement are introduced. In Section 3, the novel multi-energy load joint forecasting model and the gradient-based balance method for multi-task weights is presented. In Section 4, the preliminary settings for the case studies, including dataset segmentation, benchmark information and evaluation criteria, are provided. In Section 5, the case studies are shown, and the effectiveness of the proposed model are demonstrated. Finally, the conclusion of this paper is given in Section 6.

2. Motivation

This section introduces the proposed GNN framework and compares it with the traditional multi-task model and the traditional GNN. Regarding the key issue of how to capture coupling features of multi-energy loads, the proposed framework is clearly different from the traditional multi-task model and the traditional GNN as shown in Fig. 1.

For traditional multi-task models, a shared layer is generally required to capture the coupling features of multiple tasks. Then, each task has its own task module to capture temporal features. But this multi-task structure has a drawback. All parameters in the shared layer are affected by the goals of all sub-tasks, which may lead to insufficient representation space of temporal features of sub-tasks, and the accuracy of some tasks may decrease.

In traditional GNN, the adjacency matrix that provides the structural information between nodes is fixed as model input [28]. All structural information is stored in only one matrix, which is available to all layers [29]. Specifically, the Graph topological data can be represented as a set of nodes $\{v_i | v_i \in V\}$ and edges $\{e_{i,j} | e_{i,j} \in E\}$, in which v_i denotes a node and $e_{i,j}$ denotes the relation between v_i and v_j. Define the adjacency matrix $A = \{a_{i,j} | a_{i,j} = e_{i,j} \in \mathbb{R}^{m \times m}\}$, and $v_i \in \mathbb{R}^{1 \times n}$, then, the GNN can be represented as

$$Y = \sigma(A_{m \times m} X_{m \times n} W_{n \times k}) \tag{1}$$

where $A_{m \times m}$ denotes the adjacency matrix, $W_{n \times k}$ is the matrix of weight and $X_{m \times n} = [v_1, v_2, \ldots, v_m]^T$. The values of the adjacency matrix are typically either 0 or 1, representing the absence or presence of connections between nodes. It should be noted that the correlations among multi-energy loads are complex and time-varying. This suggests that one fixed adjacency matrix may be insufficient to represent the complex and time-varying coupling characteristics. For this problem, the decomposition-reconstruction framework [24] is a feasible solution, which addresses the issue of insufficient representation space through multi-stage modeling. However, the multi-stage approaches are often overly complex.

Therefore, the MTMs are introduced to construct GNN with multi-level coupling components, which is more concise than the decomposition-reconstruction methods. The proposed model can capture the deep coupling features of multi-energy loads at different levels of abstraction and enable the task modules to focus on the temporal features of loads. Specifically, all values in MTMs are trainable parameters in the range 0 to 1. Each layer has its task-sharing matrix at its own level of abstraction as shown in Fig. 1. The parameters in MTMs are trained along with the other parameters of the model.

For the issue of multi-task balance, the key lies in designing a balancing algorithm that is tailored to the multi-task model itself. For the design of the bottom shared layer and the specific task layer, Liu et al. [30] proposed a simple Dynamic Weight Average (DWA) method only considering the loss for each task to obtain less training time but lose important gradient information. Then, Chen et al. [31] proposed a multi-task gradient balance method to realize the simultaneous optimization of multiple tasks. However, our framework consists of the backbone and MTMs, which differs from the design of the bottom shared layer and specific task layer. This multi-task balance method cannot be directly applied.

Therefore, a Gradient-based Balance Method based on MTMs (GBMM) is proposed to adapt to the above framework. The MTMs can capture the coupling information among multi-energy loads, so the gradient of the task weights can be normalized by the average of MTMs to balance the training of tasks in real-time, thereby achieving better performance and faster speed of joint forecasting.

3. Proposed framework

This section introduces the proposed framework for multi-energy load forecasting. Firstly, the model architecture is presented and explained. Then, the gradient-based balance method for multi-task weights is proposed for the better training performance.

The joint forecasting model for multi-energy loads is presented in Fig. 2. The $input_i$ consists of the historical $load_i$ and other auxiliary data, both of which are normalized. The forecast goal is the load values for the next one hour. Each module can achieve rapid dissemination of information by short connections and Layer Normalization (LN) to promote the training of the network [32]. Multiple forecasting modules are connected by MTMs to explore the coupling features between multi-energy loads. Specifically, the backbone of the proposed framework consists of graph attention, MHA and graph convolution network,

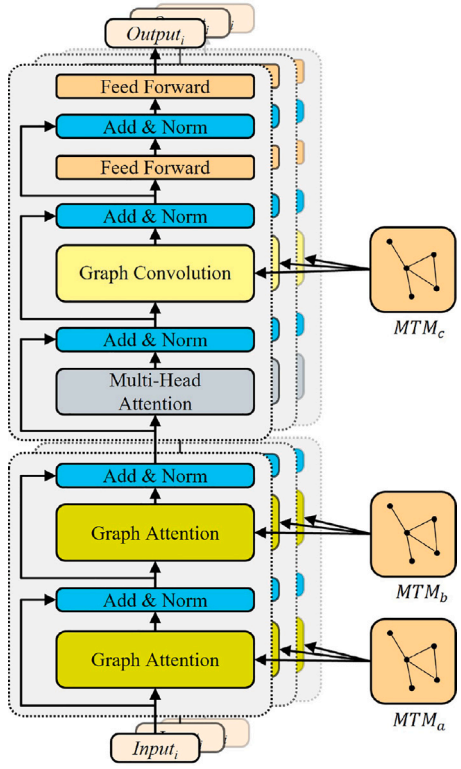


Fig. 2. Diagram of the proposed framework.

which can effectively extract the high-dimensional complex features of multi-energy loads to perform forecasting tasks. Firstly, considering that attention networks have stronger global feature extraction capabilities [33], the model preliminarily extracts data features through a two-layer graph attention with short connections. Then, MHA and graph convolution network with short connections are adopted to capture abstract features of multi-energy loads. Finally, the Feed Forward Network (FFN) with short connections is used to enhance the capacity of nonlinear features and meet the output requirements.

3.1. Graph modules with MTMs

Considering the complex coupling features of multi-energy loads, it is necessary to expand the coupling feature capacity of the model [23]. Therefore, the multi-level task-sharing matrices are proposed to realize the joint forecasting of multiple loads. Firstly, the correlation between multi-energy loads may have different characteristics at different levels of abstraction, so the model should have sufficient depth and information capabilities. Then, the coupling features can be captured separately at different levels of abstraction. Specifically, the graph attention networks and graph convolution network have their own shared matrix (MTM_a, MTM_b, MTM_c) to fit the coupling features in different abstract dimensions. Specifically, the Xavier initialization is adopted for the parameters of MTMs.

3.1.1. Graph attention

This framework has two graph attention modules, which respectively have the adjacency matrices, MTM_a and MTM_b, as shown in Fig. 2. The first graph attention is closest to the input data and thus can capture low-dimensional data features by MTM_a. The second graph attention, as an intermediate graph module, can capture the higher-dimension coupling features by MTM_b. Specifically, when graph attention aggregates nodes, the importance of different nodes varies, as shown in Fig. 3. When calculating the representation of each

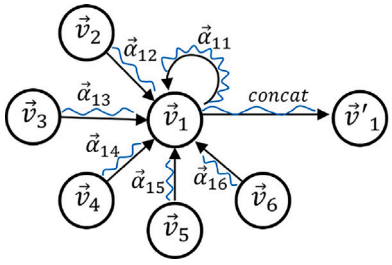


Fig. 3. Graph attention.

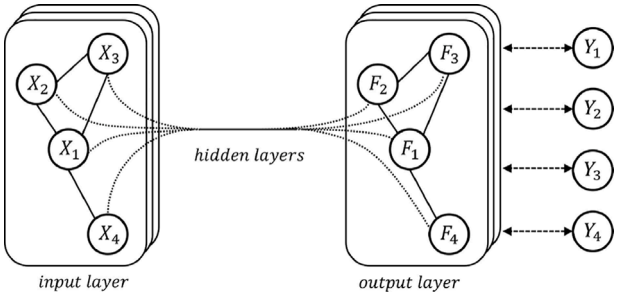


Fig. 4. Graph convolution.

node in the graph, different weights are assigned based on the features of nodes as

$$v'_i = \sigma(\sum_{j=0}^{m} \alpha_{i,j} v_i Q_{n\times n}) \tag{2}$$

where v'_i denotes the updated nodes, $\alpha_{i,j}$ denotes the attention weights, $Q_{n\times n}$ denotes the matrix of weights and $\alpha_{i,j}$ can be calculated by

$$\alpha_{i,j} = \frac{exp(LeakyReLU([v_i Q_{n\times n} \parallel v_j Q_{n\times n}]))}{\sum_{k=0}^{m} exp(LeakyReLU([v_i Q_{n\times n} \parallel v_k Q_{n\times n}]))} \tag{3}$$

where $\parallel$ denotes the concatenation operation, then, the rest of the calculations are the same as in Eq. (1). As stated above, graph attention can measure the importance of different nodes to further enhance the feature extraction ability of GNN.

3.1.2. Graph convolution

This framework has a graph convolution module with adjacency matrix MTM_c, as shown in Fig. 2. This graph convolution module is closest to the model output, so the balance of multiple tasks can calculate the weights based on MTM_c. The calculation of multi-task balance is introduced in the next subsection. Specifically, graph convolution [34] extends convolution operations from traditional data to graph data, as shown in Fig. 4. Each layer of a graph convolution network can be represented as

$$Y = f(X_{m\times n}, A_{m\times m}) \tag{4}$$

where $A_{m\times m}$ denotes the adjacency matrix in this layer, $X_{m\times n}$ denotes the input, Y denotes the output and f can be calculated by

$$f(X, A) = \sigma(D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}XW) \tag{5}$$

where W denotes the matrix of weights, I denotes the identity matrix, D denotes the degree matrix of $(A + I)$. In summary, $D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}$ denotes the symmetric normalization matrix of A.

As stated above, by introducing convolution operations, graph convolution network can enhance the ability to extract local features and obtain more complete interactive short-term coupling features. In the proposed framework, the graph convolution network is placed between the MHA and FFN to further maintain the connections among multiple tasks and to provide gradient information for the multi-task balance algorithm.

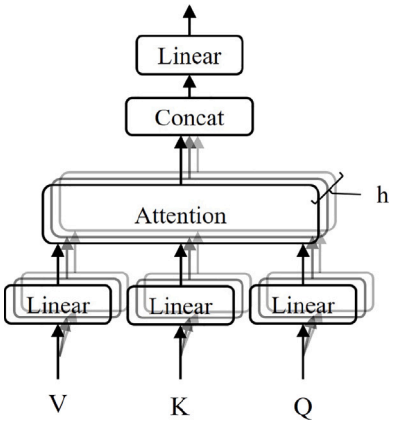


Fig. 5. Multi-Head attention.

3.2. Gradient-based balance method for multi-task weights

Each forecasting task has different optimization difficulty, so the typical fixed weight method [17] is not sufficient to balance multiple tasks. To make each forecasting task benefit from other tasks, it is necessary to explore the quantitative methods for task difficulties to balance multiple tasks in real-time during the training process. For traditional multi-task models, which have the shared layer and specific task layer, a multi-task balance method is proposed in [31], but it cannot be simply applied to the proposed GNN framework. Therefore, a modified gradient-based balance method based on MTMs named GBMM is proposed to adapt to the proposed GNN framework with MTMs.

Define the loss for task i in tth epoch as $L_i(t)$, then the overall loss of multiple tasks can be defined as

$$L(t) = \sum_{i}^{N} \omega_i(t) L_i(t) \tag{6}$$

where N denotes the number of tasks and $\omega_i(t)$ denotes the weight of task i in tth epoch. Our goal is to find the appropriate $\omega_i(t)$ to balance each task in real-time during the training process.

Different from [31], the shared matrix MTM_c are used to provide gradient information for the multi-task balance. Then, the L_2 normalization of the gradient of $\omega_i(t)L_i(t)$ with respect to MTM_c is used as an indicator of task training status, which can be calculated by

$$G_W^{(i)}(t) = \| \nabla_W \left[\omega_i(t) L_i(t)\right] \|_2 \tag{7}$$

Then, the gradient loss can be calculated by

$$L_{grad} = \sqrt{\sum_i (G_W^{(i)}(t) - \frac{1}{N} \sum_j G_W^{(j)}(t) \times r_i(t)^{\alpha})^2} \tag{8}$$

$$r_i(t) = \frac{L_i(t)}{\frac{1}{N} \sum_k L_k(t)} \tag{9}$$

where α is an hyperparameter used to set the strength of the multi-task balance. A larger value for α increases the strength of multi-task balancing but may lead to convergence issues. Conversely, a smaller α value results in insufficient multi-task balancing.

In practice, the value of α was set to 0.25, which achieves the optimal balance of our multiple tasks. Firstly, the L_{grad} is calculate and the ω_i is updated. Then, we keep the ω_i constant and update all the network parameters. The MTM_c is the task-sharing matrix closest to the model output, containing the coupling information closest to the task losses, so the gradient of the task weights can be normalized by MTM_c to balance the training of tasks in real-time, thereby improving the performance and speed of joint forecasting.

Table 2
Weather dataset.

Symbol	Description
Temperature	Air temperature at 2 m height above the earth's surface
Station pressure	Atmospheric pressure at weather station level
Relative pressure	Atmospheric pressure reduced to mean sea level
Humidity	Relative humidity at a height of 2 m above the earth's surface
Wind speed	Wind speed at a height of 10–12 m above the earth's surface
Cloud cover	Total cloud cover
Min temperature	Minimum air temperature during the past period (in 12 h)
Max temperature	Maximum air temperature during the past period (in 12 h)
Cloud height	Height of the base of the lowest clouds
Visibility	Horizontal visibility
Dewpoint	Dewpoint temperature at a height of 2 m above the earth's surface
Precipitation	Amount of precipitation

3.3. Multi-head attention

The MHA is the dot-product attention as shown in Fig. 5. Firstly, apply three row transformations with the same scale to the input data X separately to obtain Q, K, and V as intermediate variables for the next step, which can be calculated by

$$\begin{cases} Q = W_q X \\ K = W_k X \\ V = W_v X \end{cases} \tag{10}$$

where W_q, W_k and W_v are the row transformation matrices for Q, K and V, respectively.

Then, the output of $Head_i$ can be calculated as

$$\begin{aligned} Head_i &= Attention(Q, K, V) \\ &= softmax\left(\frac{(QW_i^Q)(KW_i^K)^T}{\sqrt{d_k}}\right)(VW_i^V) \end{aligned} \tag{11}$$

where W_i^Q, W_i^K and W_i^V are the column transformation matrices of Q, K and V in $Head_i$, respectively. Finally, the $MultiHead$ can be calculated by

$$MultiHead = Concat\left(Head_1, Head_2, \ldots, Head_h\right) W^M \tag{12}$$

where W^M is the output matrix. Multi-head attention can combine information from different representation subspaces, enabling the extraction of more comprehensive features and alleviating the issue of insufficient attention capacity in one single head.

4. Experiment setup

In this section, the preliminary settings for the case studies, including dataset, benchmark models and evaluation criteria, are provided.

4.1. Datasets

4.1.1. Data collection and analysis

The collected dataset includes load and weather data from 2016 to 2020, with an interval of 1 h. The multi-energy load data is collected from the Tempe campus of Arizona State University, which includes residential, academic and administration buildings. Electrical load, cold load and heat load in one week coexist in the campus as shown in Fig. 6, with cold energy carriers and heat energy carriers being cold water and hot water, respectively. In addition, the cold energy transmission system has some ice energy storage capacity. The weather data is from the nearest weather station and includes the elements shown in Table 2.

Then, the correlation of the data is analyzed based on Maximal Information Coefficient (MIC), which can be used to measure the degree

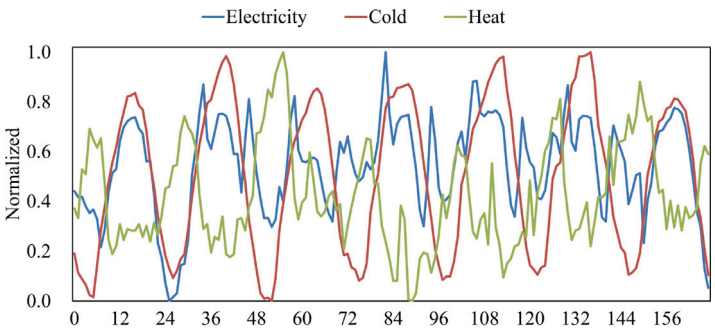


Fig. 6. Multi-energy loads in one week.

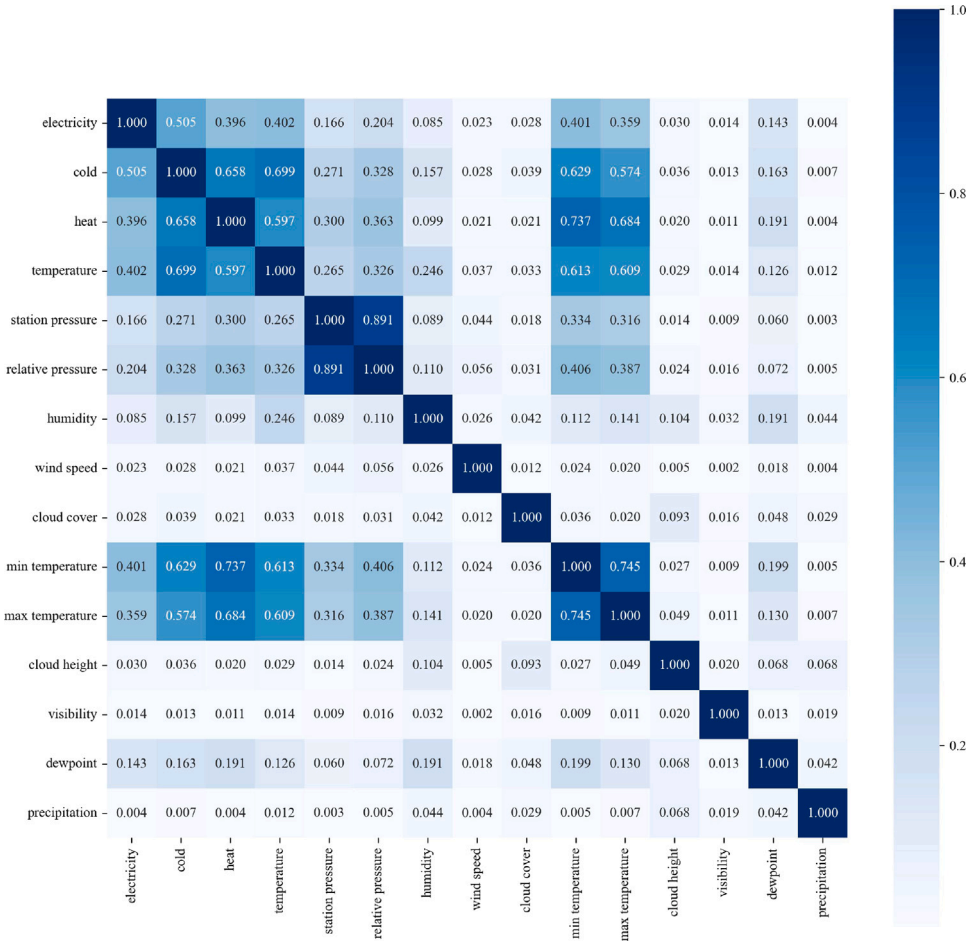


Fig. 7. MICs between multi-energy loads and weather information.

of linear or nonlinear correlation between two time series. The MICs between multi-energy loads and weather information is shown in Fig. 7. Elements with MIC values less than 0.05 are considered irrelevant, so wind speed, cloud cover, cloud height, visibility and precipitation are removed. In addition, after adding the timestamp, the feature dimension of the dataset is 15.

4.1.2. Data preprocessing

The dataset preprocessing consists of four parts: outlier detection, mean replacement, normalization and segmentation. Firstly, after obtaining the data, outliers and missing values are detected and marked. Then, the mean of the 4-point or 6-point data near the outlier is used to replace the outlier data. Next, the data of each dimension is normalized to the range of 0 to 1 respectively. Finally, the dataset is divided into a train-validation set and a test set as shown in Fig.8. To avoid the information leakage of time series data, the 5-fold time series cross-validation method is adopted to select the hyperparameters of the models.

4.2. Benchmarks and criteria

Table 3 shows the hyperparameters of the proposed and benchmark models. In this paper, Regression Forests, BiLSTM [12], RSAE [6], MultiDeT [21], CNN-LSTM [14] and TLJMD-DOE [24] are considered as the benchmark models. In addition, the single-task form of BiLSTM [12] (ST-BiLSTM) and the proposed framework are also considered as the benchmark models. Regression Forests, as a traditional ensemble forecasting method, is a commonly used benchmark model. RSAE [6] is

Table 3
Parameters of benchmarks and the proposed model.

Type	Model	Parameters
Single-Task	Random Forests	N-estimators: 100
	ST-BiLSTM	Input-FC: 64 BiLSTM-layers: [64, 64, 64, 64] Output-FC: 128
	ST-GNet	Model dimension: 32 Multi-head: 16 Hidden size: 128 Dropout: 0.1
	RSAE [6]	Num of AutoEncoder: 4 Output-FC :128
Multi-Task	MultiDeT [21]	Encoder layer: 4 Decoder layer: 4 Decoder multi-head: 8 Model dimension: 64
	MT-BiLSTM [12]	Input-FCs:64 BiLSTM-layers: [64, 64, 64, 64] Output-FCs :128
	CNN-LSTM [14]	CNN-layers:[64, 64] CNN-kernel-sizes:3, 5, 7 BiLSTM-layers: [64, 64, 64, 64]
	TLJMD-DOE [24]	GRU-layers: [512, 512], [256, 256], [256, 512] LSTM-layers: [256, 256], [256, 512], [512, 512] BiLSTM-layers: [256, 256], [256, 256], [512, 512] TCN-Filter number: 64 TCN-Filter size: 4
	MT-GNet (Proposed)	Model dimension:32 Multi-head: 16 Hidden size:128 Dropout:0.1

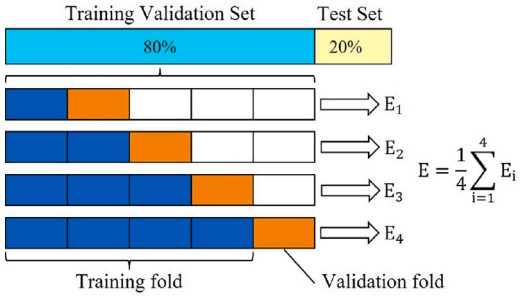


Fig. 8. Datasets.

used due to its strong uncertainty capture capability. The parameters of MultiDeT, CNN-LSTM and TLJMD-DOE are the same as those in the original paper. Specifically, the hyperparameters for all models, including the proposed method and benchmark models, are selected through the same process. Firstly, the search scope of hyperparameters is determined. Then, all models adopt the five folds time-series cross validation method on the training-validation set of Fig. 8 to adjust the hyperparameters and obtain the forecasting results on the test set. The training folds consist only of observations that occurred prior to the observation that forms the validation folds. Thus, no future observations can be used in constructing the forecast, and information leakage of time series data will not happen. The validation loss is computed by averaging over the validation folds, and this indicator was used to adjust the model hyper-parameters.

Next, take the model dimension, the head number of MHA and the neuron number of FFN as examples to illustrate the process of hyperparameter selection in Figs. 9. When the model dimension is set to 32, the model exhibits the lowest prediction error. The head number

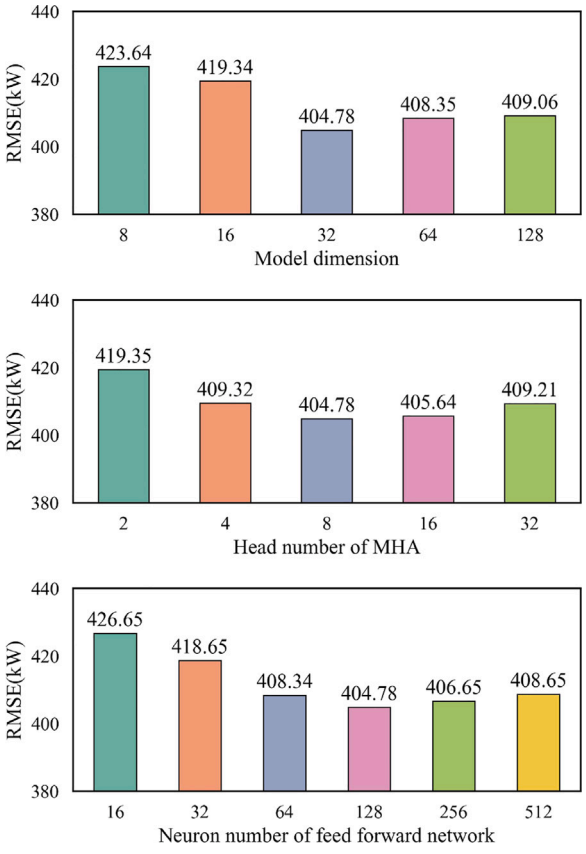


Fig. 9. RMSE under the different hyperparameters.

Table 4
RMSE, MAPE and R^2 of models.

Model	Overall			Electricity			Cold			Heat		
	RMSE (kW)	MAPE (%)	R^2 (10^{-2})	RMSE (kW)	MAPE (%)	R^2 (10^{-2})	RMSE (kW)	MAPE (%)	R^2 (10^{-2})	RMSE (kW)	MAPE (%)	R^2 (10^{-2})
Random Forests	668.08	4.18	94.35	652.07	2.37	87.60	1223.98	6.94	97.91	128.18	3.22	97.54
ST-BiLSTM	509.19	3.01	95.19	642.54	2.75	87.96	783.60	3.31	99.14	**101.44**	**2.98**	**98.46**
ST-GNet	**453.58**	**2.74**	**96.48**	**531.92**	**2.03**	**91.75**	**726.02**	**3.09**	**99.27**	102.81	3.11	98.42
RSAE [6]	593.60	3.54	95.05	628.59	2.57	88.48	1042.27	4.79	98.49	109.93	3.27	98.19
MT-BiLSTM [12]	484.29	2.97	96.06	566.63	2.25	90.64	782.21	3.68	99.15	104.04	2.99	98.38
MultiDeT [21]	463.88	2.84	95.68	603.97	2.27	89.36	682.27	3.04	99.35	105.41	3.20	98.34
CNN-BiLSTM [14]	487.05	3.11	96.05	567.32	2.26	90.61	790.99	3.95	99.13	102.85	3.12	98.42
TLJMD-DOE [24]	462.83	2.91	96.13	563.13	2.25	90.75	720.89	3.47	99.28	104.47	3.00	98.37
Proposed	**404.78**	**2.48**	**96.88**	**503.15**	**1.87**	**92.62**	**612.23**	**2.69**	**99.48**	**98.95**	**2.86**	**98.54**

Table 5
Training and testing time of the single-task models.

Model	Overall		Electricity		Cold		Heat	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Random Forests	**1 h 54 m**	**12 s**	39 m	4 s	36 m	4 s	39 m	4 s
ST-BiLSTM	6 h 52 m	36 s	2 h 18 m	12 s	2 h 16 m	12 s	2 h 18 m	12 s
ST-GNet	6 h 16 m	36 s	2 h 04 m	12 s	2 h 06 m	12 s	2 h 06 m	12 s
RSAE [6]	7 h 03 m	51 s	2 h 22 m	18 s	2 h 19 m	17 s	2 h 22 m	16 s

Table 6
Training and testing time of the multi-task models.

Model	Training	Testing
MultiDeT [21]	6 h 35 m	14 s
MT-BiLSTM [12]	7 h 21 m	14 s
CNN-BiLSTM [14]	7 h 54 m	12 s
TLJMD-DOE [24]	6 h 49 m	18 s
Proposed	**3 h 49 m**	**12 s**

of MHA has little impact on model prediction error when it is greater than or equal to 8. The optimal head number of MHA is 16. The neuron number of FFN has minimal impact on the model prediction error when it is greater than or equal to 64. The optimal neuron number of FFN is 128. Finally, the optimal parameters can be obtained by the grid search method, as shown in Table 3. For fair comparison, the training parameters of the benchmark models are consistent with those of the proposed model in this paper. In addition, the learning rate is 0.001, the batch size is 64 and the iteration number is 200. All models have the same input and targets.

In this paper, Root Mean Square Error (RMSE), Mean Absolute Percentage Error(MAPE) and Coefficient of Determination (R^2) are used to evaluate the model performance as follows.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{13}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{14}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(\bar{y} - y_i)^2} \tag{15}$$

where $\hat{y}_i$ denotes the estimated points, y_i denotes the target points, $\bar{y}$ denotes the mean of target points and N denotes the number of points.

5. Result analysis

This section introduces the forecasting results, the effect of the proposed gradient-based balance method and the ablation study.

5.1. Forecasting results

To demonstrate the advantages of the proposed model, a comparison of its performance with that of benchmark models is carried out. Table 4 and Fig. 10 show the forecasting performance of the proposed model and benchmark models on the test set. From the perspective of overall performance, the RMSE, MAPE and R^2 of the proposed model are the best, with values of 404.78, 2.48% and 96.88%, respectively. Moreover, the performance of the multi-task models is generally better than the single-task models. In the single-task models, the RMSE, MAPE and R^2 of ST-GNet are the best, with values of 453.58, 2.74% and 96.48%, respectively. Moreover, ST-GNet has better prediction accuracy than MT-BiLSTM, MultiDeT, MT-CNN-BiLSTM and TLJMD-DOE, which indicates that the proposed MTMs under the single-task structure can also capture deeper multi-energy coupling characteristics. In the multi-task models, the RMSE, MAPE and R^2 of MultiDeT are better than MT-BiLSTM and MT-CNN-BiLSTM, with values of 463.88, 2.84% and 95.68%, respectively. This indicates that the attention-based models have advantages over LSTM-type models. And the RMSE and R^2 of TLJMD-DOE are better than MultiDeT, with values of 462.83 and 96.13%, respectively. This indicates that the ensemble models can take advantage of the strengths of various models. As stated above, the proposed framework can capture the coupling characteristics of multi-energy loads by MTMs, which better matches the inherent coupling features, so that the forecasting of cold load can also benefit from the forecasting of electricity and heat loads, and thus improve the forecasting accuracy of all loads.

Table 5 and 6 shows the training and testing time for the proposed and benchmark models. In the single-task models, Random Forests has the shortest training time due to its distinctive machine learning architecture. Furthermore, the training time of most neural network models is more than 6 h. However, the training time of the proposed model is only 3 h and 49 min. This is because MTMs can capture coupling features, enabling the main body of the model to focus on temporal features. Therefore, the proposed GNN architecture with MTMs has higher computational efficiency.

To better demonstrate the forecasting performance advantages of the proposed model, the Diebold–Mariano (DM) test is carried out. Table 7 shows the DM values and p value for the benchmark models compared with the proposed model. The DM test can illustrate the performance gap and the confidence p value of the two forecasting sequences compared with the true sequences. A negative DM value indicates that the forecasting of this model is worse than that of the proposed model. If the p value is less than 0.05, it indicates that the DM value is reliable. As shown in Table 7, all p values are less than 0.05, which indicates that the DM values can reliably represent

Table 7
Diebold–Mariano test compared with the proposed model.

Model	Electricity		Cold		Heat	
	DM	p	DM	p	DM	p
Random Forests	−14.32	1.78e−45	−9.57	1.80e−21	−5.14	2.82e−07
ST-BiLSTM	−20.37	3.52e−88	−10.64	3.98e−26	**−2.78**	5.53e−03
ST-GNet	**−7.98**	1.86e−15	**−9.87**	9.58e−23	−6.29	3.56e−10
RSAE [6]	−16.26	8.76e−58	−20.03	1.84e−85	−9.03	3.28e−19
MT-BiLSTM [12]	−14.75	4.37e−48	−13.62	2.23e−41	**−3.80**	1.50e−04
MultiDeT [21]	**−13.02**	6.43e−38	**−7.01**	3.01e−12	−7.04	2.20e−12
CNN-BiLSTM [14]	−14.56	5.97e−47	−14.18	1.15e−44	−5.20	2.07e−07
TLJMD-DOE [24]	−14.46	2.52e−46	−12.56	1.37e−35	−4.32	1.57e−05

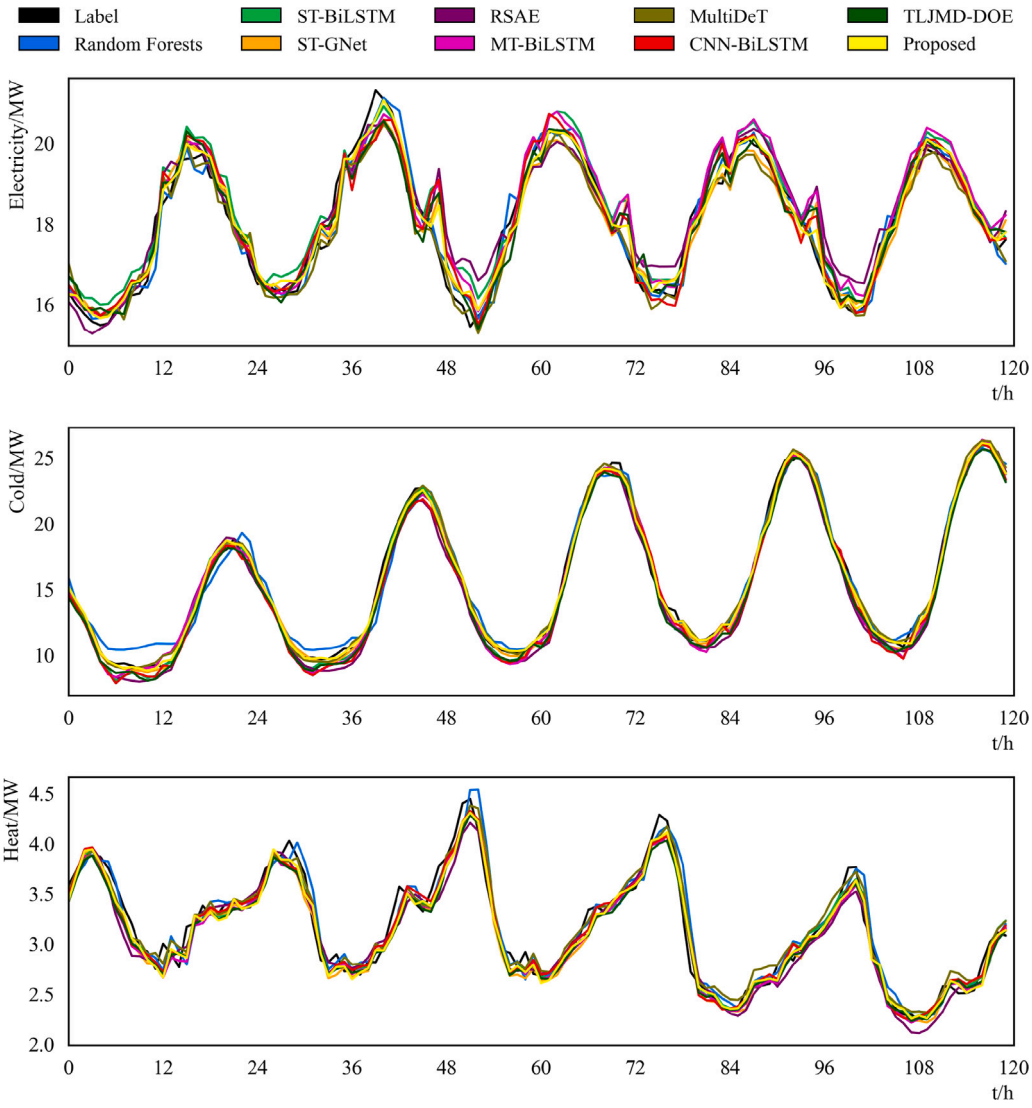


Fig. 10. Multi-energy load forecasting results.

the forecasting gaps. Then, all the DM values were negative, which indicates that the proposed model has the best forecasting performance.

To verify the robustness of the model, the proposed model was trained five times, and the results were summarized into a box plot as shown in Fig. 11. The proposed model has good robustness against the data uncertainty and model uncertainty. As a multi-task model, MT-BiLSTM, has bigger fluctuations of the RMSE, MAPE and R^2 than the proposed model, which indicates that not all multi-task structures can benefit all tasks, but those that are suitable for the current data features are.

5.2. Effect of the gradient-based balance method

To verify the effectiveness of the proposed gradient-based balance method, a comparison is set up between the proposed GBMM method, DWA method [30] and Fixed Weight (FW) method. Regarding the FW method, the weights of electricity load, cold load and heat load are fixed at 0.5, 0.3 and 0.2, respectively.

Table 8 shows the effect of the proposed GBMM method. The multi-task balance method can enhance the forecasting accuracy of multi-task models. Specifically, the proposed GBMM method has significant forecasting advantages. In terms of cold load forecasting, the performance

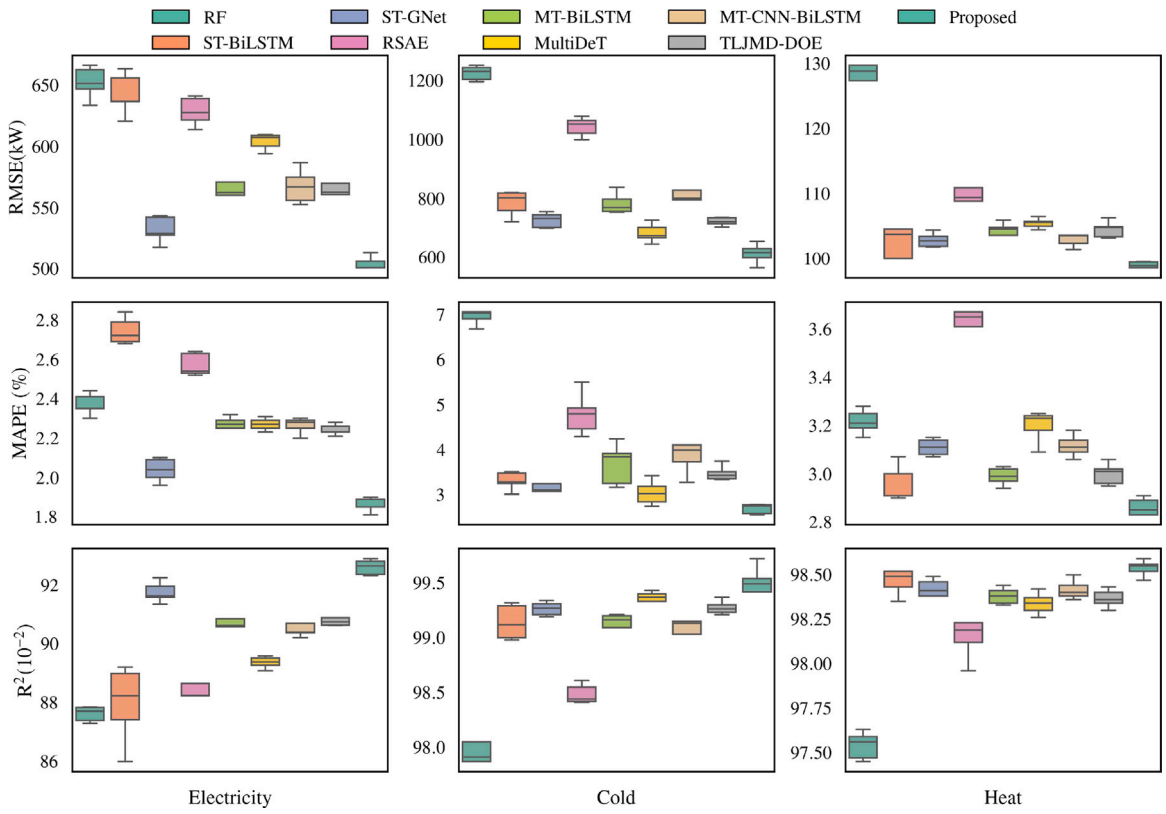


Fig. 11. Distributions of RMSE, MAPE and R^2 of multi-energy loads.

Table 8
Performance of the FW method, the DWA method and the proposed GBMM method.

Method	Overall			Electricity			Cold			Heat		
	RMSE (kW)	MAPE (%)	R^2 (10^{-2})	RMSE (kW)	MAPE (%)	R^2 (10^{-2})	RMSE (kW)	MAPE (%)	R^2 (10^{-2})	RMSE (kW)	MAPE (%)	R^2 (10^{-2})
FW	407.93	2.53	96.69	509.17	1.89	92.35	615.26	2.80	99.32	99.35	2.91	98.41
DWA	406.95	2.50	96.75	508.54	1.89	92.39	613.06	2.73	99.40	99.26	2.89	98.46
GBMM	**404.78**	**2.48**	**96.88**	**503.15**	**1.87**	**92.62**	**612.23**	**2.69**	**99.48**	**98.95**	**2.86**	**98.54**

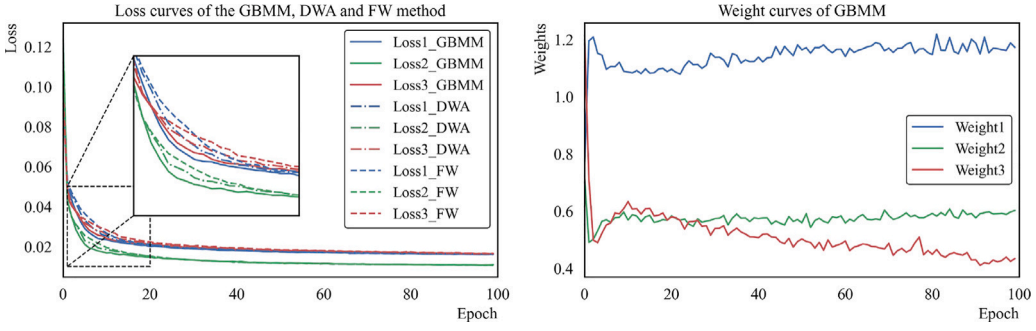


Fig. 12. Losses and weights of multi-task balance methods.

of the two balance methods is not significantly different. It is speculated that this is because the cold load has strong regularity and has a certain ice storage capacity to smooth the load curve. Fig. 12 shows that the proposed GBMM method can balance multiple forecasting tasks in real-time and accelerate the training of models. The weights of the DWA method have no practical significance and thus are not presented. The weight of the electricity load rapidly increases and gradually stabilizes at 1.2 over the training epoch. The weights of cold load and heat load rapidly decrease and gradually stabilize at 0.6 and 0.4 over the training epoch. As stated above, the proposed multi-task balance method can improve the forecasting performance.

5.3. Ablation study

To verify the effectiveness of the graph attention, graph convolution and MHA, five cases are set up in this subsection as follows.

Case 1-NoGA1: Framework without the first graph attention.

Case 2-NoGA2: Framework without the second graph attention.

Case 3-NoGA: Framework without both graph attention.

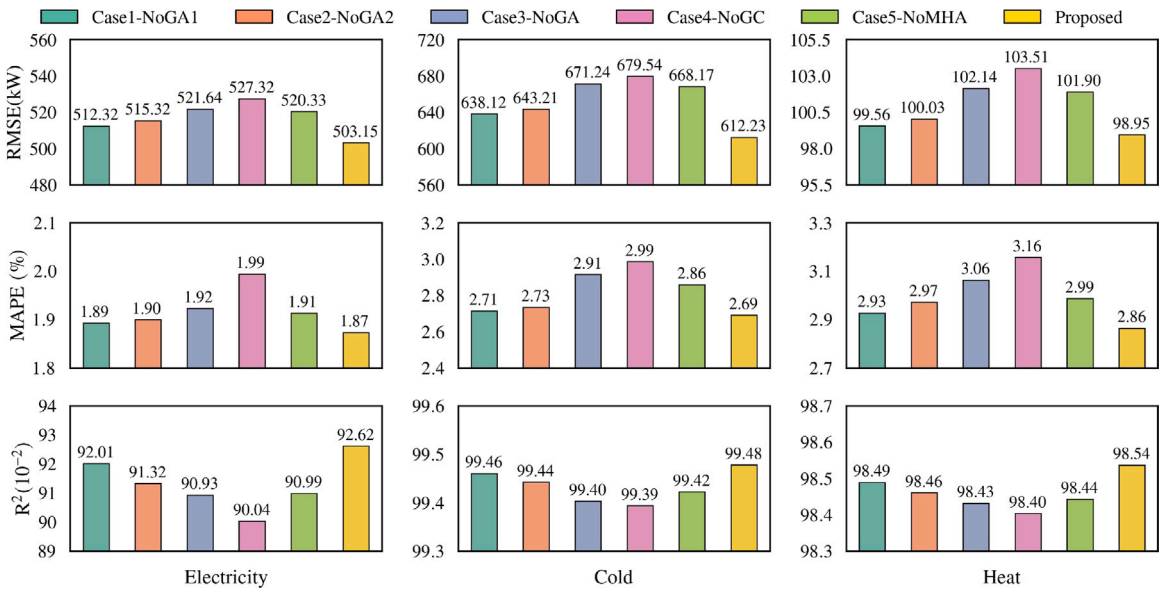


Fig. 13. RMSE, MAPE and R^2 of five cases.

Case 4-NoGC: Framework without graph convolution.

Case 5-NoMHA: Framework without MHA.

In the different cases, all models have been trained for 200 epochs with the train-validation set, and the forecasting results on the test set are shown in Figs. 13. The comparison of the results of the five cases demonstrates the effectiveness of graph attention, graph convolution and MHA in the proposed framework.

The comparison between Case 1 and Case 2 shows that the low-dimensional coupling features captured by the first graph attention are more important than the high-dimensional features captured by the second graph attention for forecasting performance. In Case 4, due to the lack of graph convolution, the proposed GBMM algorithm uses the adjacency matrix of the second graph attention as the base. The accuracy of Case 4 is lower than that of Case 3, which indicates that graph convolution has a more significant impact on model performance than graph attention. The accuracy of Case 5 indicates that the impact of MHA on the forecasting performance is slightly smaller than that of the graph attention modules. The proposed model has higher accuracy compared with other cases, which indicates that all load forecasting benefit from the joint forecasting of multiple loads. Specifically, the forecasting of the cold load and heat load of the proposed model is improved the most compared with other cases. This suggests that the cold load and heat load benefit the most from the joint forecasting of multiple loads.

6. Conclusion

This paper proposes a novel multi-energy load forecasting framework, which first introduces the multi-level task-sharing matrices, equipping GNN with multi-level coupling components to capture deep coupling features between different types of loads at different levels of abstraction. The graph attention and graph convolution further expand the representational space of coupling features, so that Multi-Head Attention (MHA) can focus on capturing the temporal characteristics of individual loads. Then, a gradient-based balance method for multiple tasks is proposed for the above framework to achieve the best performance of joint forecasting. Case studies run on a publicly available dataset demonstrate that the proposed model has superior performance than benchmark models for the forecasting of multi-energy loads.

Optimal multi-task structures and balance methods to best match the inherent coupling regularities of multi-energy loads will be our next topic of interest for our research.

CRediT authorship contribution statement

Chen Wang: Writing – original draft, Software, Methodology, Conceptualization. **Ying Wang:** Writing – review & editing, Formal analysis, Conceptualization. **Enrico Zio:** Writing – review & editing, Supervision. **Kaifeng Zhang:** Supervision.

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.

Acknowledgments

This work is funded by the "Zhishan Young Scholars" Support Program of Southeast University.

Data availability

Data will be made available on request.

References

[1] Ahmad A, Xiao X, Mo H, Dong D. Tftformer: A novel transformer based model for short-term load forecasting. Int J Electr Power Energy Syst 2025;166:110549.
[2] Wang Y, Zhang K, Qu K. Segmented real-time dispatch model and stochastic robust optimization for power-gas integrated systems with wind power uncertainty. J Mod Power Syst Clean Energy 2023;11:1480–93.
[3] Li K, Duan P, Cao X, Cheng Y, Zhao B, Xue Q, et al. A multi-energy load forecasting method based on complementary ensemble empirical model decomposition and composite evaluation factor reconstruction. Appl Energy 2024;365:123283.
[4] Kim N, Park H, Lee J, Choi JK. Short-term electrical load forecasting with multidimensional feature extraction. IEEE Trans Smart Grid 2022;13:2999–3013.
[5] Li J, Wei S, Dai W. Combination of manifold learning and deep learning algorithms for mid-term electrical load forecasting. IEEE Trans Neural Networks Learn Syst 2023;34:2584–93.
[6] Khodayar M, Kaynak O, Khodayar ME. Rough deep neural architecture for short-term wind speed forecasting. IEEE Trans Ind Informatics 2017;13:2770–9.
[7] Lin W, Wu D, Boulet B. Spatial–temporal residential short-term load forecasting via graph neural networks. IEEE Trans Smart Grid 2021;12:5373–84.
[8] Zhao P, Hu W, Cao D, Zhang Z, Liao W, Chen Z, et al. Enhancing multivariate, multi-step residential load forecasting with spatiotemporal graph attention-enabled transformer. Int J Electr Power Energy Syst 2024;160:110074.

[9] Jiang L, Wang X, Li W, Wang L, Yin X, Jia L. Hybrid multitask multi-information fusion deep learning for household short-term load forecasting. IEEE Trans Smart Grid 2021;12:5362–72.
[10] Qin J, Zhang Y, Fan S, Hu X, Huang Y, Lu Z, et al. Multi-task short-term reactive and active load forecasting method based on attention-lstm model. Int J Electr Power Energy Syst 2022;135:107517.
[11] Zhang Y, Cui Q, Shi L, Pan J, Li J. Ppenergynet: Privacy-preserving multi-energy load forecasting in energy internet considering energy coupling. IEEE Trans Power Syst 2024;39:6235–48.
[12] Guo Y, Li Y, Qiao X, Zhang Z, Zhou W, Mei Y, et al. Bilstm multitask learning-based combined load forecasting considering the loads coupling relationship for multienergy system. IEEE Trans Smart Grid 2022;13:3481–92.
[13] Niu D, Yu M, Sun L, Gao T, Wang K. Short-term multi-energy load forecasting for integrated energy systems based on cnn-bigru optimized by attention mechanism. Appl Energy 2022;313:118801.
[14] Li K, Mu Y, Yang F, Wang H, Yan Y, Zhang C. A novel short-term multi-energy load forecasting method for integrated energy system based on feature separation-fusion technology and improved cnn. Appl Energy 2023;351:121823.
[15] Shi J, Teh J. Load forecasting for regional integrated energy system based on complementary ensemble empirical mode decomposition and multi-model fusion. Appl Energy 2024;353:122146.
[16] Song C, Yang H, Cai J, Yang P, Bao H, Xu K, Meng X-B. Multi-energy load forecasting via hierarchical multi-task learning and spatiotemporal attention. Appl Energy 2024;373:123788.
[17] Wang L, Tan M, Chen J, Liao C. Multi-task learning based multi-energy load prediction in integrated energy system. Appl Intell 2023;53:10273–89.
[18] Zhou B, Meng Y, Huang W, Wang H, Deng L, Huang S, Wei J. Multi-energy net load forecasting for integrated local energy systems with heterogeneous prosumers. Int J Electr Power Energy Syst 2021;126:106542.
[19] Xuan W, Shouxiang W, Qianyu Z, Shaomin W, Liwei F. A multi-energy load prediction model based on deep multi-task learning and ensemble approach for regional integrated energy systems. Int J Electr Power Energy Syst 2021;126:106583.
[20] Li C, Li G, Wang K, Han B. A multi-energy load forecasting method based on parallel architecture cnn-gru and transfer learning for data deficient integrated energy systems. Energy 2022;259:124967.
[21] Wang C, Wang Y, Ding Z, Zheng T, Hu J, Zhang K. A transformer-based method of multienergy load forecasting in integrated energy system. IEEE Trans Smart Grid 2022;13:2703–14.
[22] Wang C, Wang Y, Ding Z, Zhang K. Probabilistic multi-energy load forecasting for integrated energy system based on bayesian transformer network. IEEE Trans Smart Grid 2024;15:1495–508.
[23] Zhuang W, Fan J, Xia M, Zhu K. A multi-scale spatial–temporal graph neural network-based method of multienergy load forecasting in integrated energy system. IEEE Trans Smart Grid 2024;15:2652–66.
[24] Lin Z, Lin T, Li J, Li C. A novel short-term multi-energy load forecasting method for integrated energy system based on two-layer joint modal decomposition and dynamic optimal ensemble learning. Appl Energy 2025;378:124798.
[25] Hu J, Duan P, Cao X, Xue Q, Zhao B, Zhao X, et al. A multi-energy load forecasting method based on the mixture-of-experts model and dynamic multilevel attention mechanism. Energy 2025;324:135947.
[26] Qu Z, Meng Y, Hou X, Chi R, Ai Y, Wu Z. Integrated energy short-term multivariate load forecasting based on patchtst secondary decoupling reconstruction for progressive layered extraction multi-task learning network. Expert Syst Appl 2025;269:126446.
[27] Wang S, Wang S, Chen H, Gu Q. Multi-energy load forecasting for regional integrated energy systems considering temporal dynamic and coupling characteristics. Energy 2020;195:116964.
[28] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. IEEE Trans Neural Netw 2009;20:61–80.
[29] Simeunović J, Schubnel B, Alet P-J, Carrillo RE. Spatio-temporal graph neural networks for multi-site pv power forecasting. IEEE Trans Sustain Energy 2021;13:1210–20.
[30] Liu S, Johns E, Davison AJ. End-to-end multi-task learning with attention. In: 2019 IEEE/CVF conference on computer vision and pattern recognition. CVPR; 2019, p. 1871–80.
[31] Chen Z, Badrinarayanan V, Lee C-Y, Rabinovich A. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: Proceedings of the 35th international conference on machine learning, volume 80 of proceedings of machine learning research. PMLR; 2018, p. 794–803.
[32] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst 2017;30.
[33] Jiang N, Gao L, Duan F, Wen J, Wan T, Chen H. San: Attention-based social aggregation neural networks for recommendation system. Int J Intell Syst 2022;37:3373–93.
[34] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. 2016, arXiv preprint arXiv:1609.02907.