

Transformer for Nonintrusive Load Monitoring: Complexity Reduction and Transferability

Lingxiao Wang¹, Shuwen Mao¹, *Fellow, IEEE*, and R. Mark Nelms¹, *Fellow, IEEE*

Abstract—Nonintrusive load monitoring (NILM) is to obtain individual appliance’s electricity consumption from aggregated smart meter data. In this article, we propose a middle window transformer model, termed *Midformer*, for NILM. Existing models are limited by high computational complexity, dependency on data, and poor transferability. In *Midformer*, we first exploit patchwise embedding to shorten the input length, and then reduce the size of queries in the attention layer by only using global attention on a few selected input locations at the center of the window to capture the global context. The cyclically shifted window technique is used to preserve connection across patches. We also follow the pretraining and fine-tuning paradigm to relieve the dependency on data, reduce the computation in modeling training, and enhance transferability of the model to unknown tasks and domains. Our experimental study using two real-world data sets demonstrates the superior performance and transferability of *Midformer* over three baseline models.

Index Terms—Attention, nonintrusive load monitoring (NILM), smart home, transferability, transformer.

I. INTRODUCTION

THE RECENT advances in the Internet of Things (IoT) allow the deployment of uniquely identifiable objects that are organized in an Internet-like structure to enable smart homes to monitor, control, and manage house appliances [1]. The communication paths constructed by the IoT integrate smart meters, home appliances, and renewable energy, in a home energy management system (HEMSs) [2], [3]. With more and more IoT-enabled technologies being developed and deployed, the HEMS system will become more sustainable, more resilient, and more energy efficient [4].

One important application of the IoT in HEMSs is load monitoring. The built-in sensors in appliances provide individual appliance’s energy consumption information to the HEMS in real time, which can be analyzed to optimize the energy usage and achieve energy savings. However, there are several practical issues that need to be addressed. First, electrical appliances typically last up to decades. As a result, a household typically include both old and new generations of appliances. The legacy appliances may not be equipped with smart sensors

and their electricity consumption data are usually hard to measure. Second, the cost of installing sensors to legacy appliances could be high, including both the sensor and installation cost, as well as the power usage cost. Third, consumers are more and more concerned about their privacy; they may not be willing to share the information about their appliances’ power consumption.

Nonintrusive load monitoring (NILM), which is to identify individual appliance’s electricity consumption from the given aggregated smart meter data, provides a useful solution to the above problems [5]. Recently, deep neural networks (DNNs), such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been shown effective to address the NILM problem. Since 2017, the Transformer [6] and its variants have dominated the field of natural language process (NLP), achieving superior performance for tasks such as language translation, text analytics, smart assistants, and so on. This is largely due to Transformer’s capability of using the attention mechanism to capture the long-range dependency in sequential data. For computer vision (CV) tasks, the vision transformer (ViT) [7] has been shown to outperform the popular CNN model. In our recent work [8], a deep spatiotemporal attention approach was developed to forecast the temperature of stored grain using meteorological data. Such successes in NLP, CV, and other fields have attracted researchers to investigate Transformer’s application to the NILM problem.

Although some recent preliminary studies have demonstrated the high potential of the Transformer for NILM [9], [10], there are still many challenges remain to be addressed. First is the tradeoff between computational complexity and the ability to track long-range dependency in energy consumption data, which usually contains rich daily, seasonal, and even annual patterns. The self-attention mechanism is the core of Transformer, which has a quadratic time complexity with regard to the input sequence length [11]. Low-complexity models are thus desirable to allow longer input sequences. Second is the dependency on data. Like most deep learning (DL) models, the Transformer requires a large amount of high quality labeled data for training, specifically, each individual appliance’s power consumption data. The cost of data collection, e.g., submetering, could be high. In addition, many users are unwilling to share their appliance’s information due to concern of privacy breach. Third is the generalization or transferability of the well-trained Transformer model. The existing Transformer-based NILM methods are trained and tested on the same data set, or assume the training and testing sets share similar data

Manuscript received 20 November 2021; revised 17 February 2022; accepted 27 March 2022. Date of publication 30 March 2022; date of current version 23 September 2022. This work was supported in part by NSF under Grant DMS-1736470 and Grant CNS-2107190, and in part by the Wireless Engineering Research and Education Center at Auburn University, Auburn, AL, USA. (Corresponding author: Shuwen Mao.)

The authors are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: lzw0039@auburn.edu; smao@ieee.org; nelmsrm@auburn.edu).

Digital Object Identifier 10.1109/JIOT.2022.3163347

distribution. The transferability of the models have not been fully investigated, including testing across different appliances and/or across different data sets. NILM models with strong transferability are useful to achieve accurate predictions for different, unseen houses, different models or brands of appliances, various aging degrees of electronic circuits, and different residents daily habits and usage behavior [2].

In this article, we propose a middle window transformer model, termed Midformer, for NILM, which incorporates several novel designs and follows the pretraining and fine-tuning paradigm to address the above problems. To deal with the computational complexity issue, Midformer is designed as a more efficient Transformer variant tailored to the characteristics of the NILM problem. Specifically, we utilize patchwise attention in Midformer, which reduces the input length compared to pointwise attention used in existing models [9], [10]. We further apply the cyclically shifted window technique to increase the receptive field. The drawback of patchwise attention is that it ignores the connection across patches. In Midformer, we feed both cyclically shifted input and the original input into the attention layer to preserve the connection across patches. To reduce computation, we only calculate full attention using the middle range of the input, instead of using the entire input. This allows Midformer to focus on the middle range of the input and achieve a linear time complexity with respect to the input length (i.e., the window size).

To address the transferability issue and reduce the dependency on data, we follow the pretraining and fine-tuning paradigm. First, we pretrain multiple transformers (for different appliances) by using one data set. Then, we test the performance of the trained models on unseen data in the same data set. Next, we examine the relationship among different appliances, i.e., could a model pretrained using one appliance's data in a house be used to predict the power usage of another appliance in another house? D'Incecco *et al.* [12] used the model learned from washing machine data to predict the power consumption of other appliances. In this article, we obtain the pretrained model (including CNN, RNN, Transformer, and the proposed Midformer) for five appliances (including kettle, dishwasher, fridge, washing machine, and microwave). We then fine-tune and test the pretrained model on a different data set including the same and different appliances' data. With this approach, models do not need to be retrained from scratch for unknown houses and unseen appliances, and can quickly adapt to new tasks with few-shot fine-tuning due to the well initialized parameters in the pretrained models. This way, the computation in modeling training can be reduced and the dependency on data can be relieved.

We evaluate the performance of the proposed Midformer model using two real-world data sets and compare it with three baseline models, including CNN, RNN, and Transformer. Our experimental study demonstrates the superior performance and great transferability of the proposed Midformer model for NILM problems over the state-of-the-art baseline models.

We organize the remainder of this article as follows. We introduce the related work in Section II. In Section III, we formulate the NILM problem and introduces the preliminaries of Transformers. In Section IV, we present the proposed

transformer method Midformer. We present the data sets and experiment setup, and discuss the experimental study in Section V. Finally, Section VI concludes this article.

II. RELATED WORK

A. Nonintrusive Load Monitoring Models

In the literature, many prior studies have developed approaches for solving the NILM problem, which can be mainly divided into two categories: 1) unsupervised learning and 2) supervised learning methods. In this section, we will briefly introduce the existing solutions for NILM; more detailed reviews of the different approaches applied to solving NILM can be found in [5], [13], and [14].

1) *Unsupervised Learning*: Unsupervised learning has the unique strength of not requiring labeled data. The additive factorial hidden Markov model (AFHMM) is one of the most widely used unsupervised learning approaches for NILM [15]–[18], which converts time series data into hidden-Markov models and Bayesian models to infer the possible states of different appliances. Another method of unsupervised learning approach is the graph signal processing (GSP)-based method, which has also been shown to be quite effective for NILM [19], [20]. The main drawbacks of these methods is that the prior domain knowledge needs to be provided, and such schemes may not perform well for solving problems that have a large number of appliances [12].

2) *Supervised Learning*: Supervised learning aims to learn a function, which maps an input to an output, from given input–output examples, i.e., labeled data. In the literature, various supervised learning methods have been applied to solving the NILM problem, such as support vector machine [21], decision tree [22], K -nearest neighbors (k-NN) [23], and so forth. Recent works in this area demonstrate the promise of entirely DL approaches, such as CNNs [24]–[27], long short-term memory (LSTM) or its variant gated recurrent units (GRUs) [28]–[31], and denoising autoencoder [32], [33]. The main limitation of supervised learning (machine learning) method is that it requires large amounts of high-quality training data. Such approaches usually require high computational power and storage capacity.

B. Transfer Learning for NILM

Most of the approaches applied to the NILM problem are carried out on the same data domain, which means the model is trained and tested using the same appliance's data in the same data set. Very few previous studies have addressed the study of generalizability of the NILM models, also referred to as the transferability of the pretrained models. For example, Murray *et al.* [34] trained two different networks based on CNNs and RNNs, respectively, by using one of the three data sets and verify the models' transferability as well as generalization through the other data sets. However, the stability of the trained models is unsatisfactory due to the different data distributions in different databases, which lead to the poor domain adaptation performance.

To address this problem, D'Incecco *et al.* [12] pretrained their sequence-to-point (seq2point) learning model using the

washing machine data in one specific data set, and then tested their pretrained model on data of different appliances in different data sets. Fine-tuning, which is to train the pretrained model using a small amount of examples from the testing data set [2], was applied to adapt the pretrained model to the difference between the different training and testing domains. However, the distribution of data used in fine-tuning was quite different from that of the tested house data, which led to the *negative transfer* effect. The generative adversarial networks (GANs) model has been applied to address the domain adaptation problem in NILM as well [35], [36], which was to train the feature generator and the domain discriminator in the adversarial manner. The limitation of this method is that training GANs requires finding a Nash equilibrium of a nonconvex game with continuous high-dimensional parameters, which could fail to converge [37], [38]. Our previous work [2] developed a metalearning-based approach and an ensemble-learning-based approach that require fewer new data for adaptation, and can quickly adapt to new NILM tasks. However, we only explored the transferability between different data sets of same appliance in [2].

C. Transformer-Based NILM Models

Motivated by the success of the Transformer architecture in many domains, most importantly in NLP [6], the self-attention¹ based Transformer has recently been proposed for NILM. The recent works [9], [10] both applied the attention mechanism to the feature maps extracted by CNNs to solve NILM tasks. The main drawback of these preliminary studies is that the computational complexity of self-attention grows quadratically with window (i.e., input) size, which could become a serious issue if the fixed window size is large. Moreover, the generalization performance of these models have not been verified through different data sets or appliances.

III. PROBLEM STATEMENT

A. NILM Problem

Consider a given collection of J time series $\{y_1(t), y_2(t), \dots, y_J(t)\}_{t=1}^T$ that record the energy consumption of J appliances in a house over a period of time T ; and $\{y_j(t)\}_{t=1}^T$ represents the power consumption of the j th appliance in the house. The aggregate power consumption $x(t)$ of the house at time t is calculated as follows:

$$x(t) = \sum_{j=1}^J y_j(t) + e(t) \quad (1)$$

where $e(t)$ is the measurement noise at time t . The NILM problem is to estimate the power consumption of an individual appliance from the given aggregate power consumption of the entire house. It is also called energy disaggregation since the goal is to separate the energy consumption measured at the aggregate level to that of individual appliances. It is non-intrusive since only the aggregate measurement is needed; and there is no need for submetering.

¹An attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence [6].”

In NILM algorithms, to better handle the long time series data, usually a sliding/rolling window setting is adopted over the time series with a fixed window size, denoted by W , where the sliding/rolling step size is one. Rather than predicting a full window size of outputs, the NILM models often target at one single time instance (e.g., the middle point of the window) to avoid redundant computation. This approach is termed s2p learning [24]. Therefore, given input data of total power consumption measurements over a window of size W , i.e., $\tilde{\mathbf{x}} = \{x(1), x(2), \dots, x(W)\}$, the learning algorithm will compute output $\tilde{y}_j(\lceil W/2 \rceil)$, for all j .

B. Transformer and Multihead Self-Attention Mechanism

The Transformer model is based on the attention mechanism to significantly enhance the performance of DL, which computes the representation of a sequence by attending to information at different positions from different representation subspaces [6], [39]. The main idea of this mechanism is to learn an alignment between each element in the sequence and others to decide which part of the sequence should be paid attention to [40].

For a given input sequence $\mathbf{I} \in \mathbb{R}^{W \times d_{\text{model}}}$, where d_{model} is the dimension of each data sample (i.e., length of the encoding vector), self-attention first transforms the input sequence into three matrices with three learnable weights. These three matrices are called queries, keys, and values, respectively, and they have the same depth of dimension d . Next, the scaled dot-product is computed, which is given by

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \quad (2)$$

where $\mathbf{Q} = \mathbf{I}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{I}\mathbf{W}^K$, and $\mathbf{V} = \mathbf{I}\mathbf{W}^V$, and $\mathbf{W}^Q \in \mathbb{R}^{d_{\text{model}} \times d}$, $\mathbf{W}^K \in \mathbb{R}^{d_{\text{model}} \times d}$, and $\mathbf{W}^V \in \mathbb{R}^{d_{\text{model}} \times d}$ are all trainable parameters that are used to map the input \mathbf{I} into the three matrices \mathbf{Q} , \mathbf{K} , and \mathbf{V} . The attention function (2) is similar to nonlocal means, which can be described as mapping a query and a set of key-value pairs to an output [6]. The weighted sum of the values is computed as the output of attention, where the weight is determined by the softmax score of the query with the corresponding key.

In the Transformer model, the attention processor is also called the attention head. Multihead self-attention computes the self-attention score function describe in (2) on H different linear projections of queries, keys, and values in parallel. Then, the results are concatenated as follows:

$$\text{MultiHead}(\mathbf{I}) = \text{Concat}\left(\sum_{i=1}^H \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)\right) \quad (3)$$

where $\mathbf{Q}_i = \mathbf{I}\mathbf{W}_i^Q$, $\mathbf{K}_i = \mathbf{I}\mathbf{W}_i^K$, and $\mathbf{V}_i = \mathbf{I}\mathbf{W}_i^V$. The dimension of the learning parameters \mathbf{W}_i^Q , \mathbf{W}_i^K , and \mathbf{W}_i^V is $d_{\text{model}} \times d_i$, where $d_i = d/H$. By combining several similar attention results, the attention will have stronger power of discrimination.

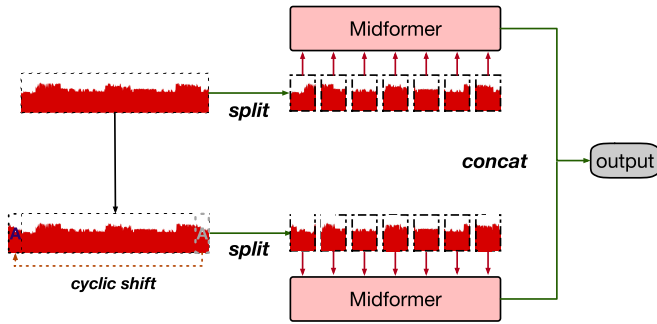


Fig. 1. Architecture of the proposed transformer-based approach, Midformer, to NILM.

IV. PROPOSED MIDDLE WINDOW TRANSFORMER APPROACH

In this section, we introduce our proposed Transformer-based approach for NILM problems, which is termed middle window transformer (Midformer). Fig. 1 illustrates the overall architecture, which consists of four main parts, including: 1) patch splitting and initializing; 2) cyclic shift window; 3) Transformer layers; and 4) concatenation. Our intuition of designing this approach is to utilize the Transformer's attention ability to model the long-range dependency in the energy consumption data, while reducing the computational cost.

The existing methods [9], [10] exploit the attention mechanism for NILM by combining CNNs with forms of self-attention. They first extract the feature map from input data by using convolutional layers. The extracted feature map is then fed into the Transformer layers. They both adopt global full self-attention in their models, which has a computational complexity that is quadratic to the size of the feature map. For efficient modeling and computation, we leverage the technique proposed in the ViT for image classification tasks [7], which reduces the context length by partitioning images into small patches and using the patches as input to the Transformer layers. A comparison of the existing approach and that adopted in this article is presented in Fig. 2. In particular, Fig. 2(a) shows the original pointwise attention projection method used in existing NILM works [9], [10], while Fig. 2(b) illustrates the patchwise attention projection method adopted in this article. As shown in Fig. 2(a), when creating the attention matrix, the input will first be mapped into a space of depth d , which will then be used by the attention mechanism to calculate the attention matrix. The length and width of the attention matrix are the same as the depth of the space. From the comparison figure, we can visually see that the patchwise attention incurs significantly less computation than the pointwise attention as the dimension of the attention matrix becomes much smaller.

A. Patch Splitting and Initialization

In the proposed Midformer model, the input $\mathbf{I} \in \mathbb{R}^{W \times d_{\text{model}}}$ is first split into a sequence of nonoverlapping patches of fixed-size $\{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_k\}$, where $\mathbf{I}_i \in \mathbb{R}^{W/k \times d}$, for $1 \leq i \leq k$, and k is the number of patches. Each patch contains W/k samples, and is fed into a neuron network to be projected into

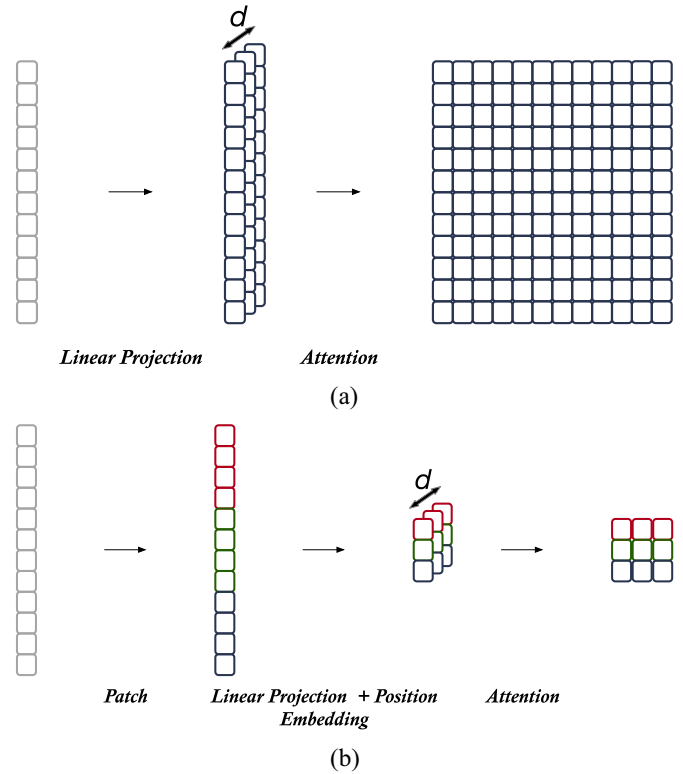


Fig. 2. Comparing the pointwise and the patchwise attention pattern. (a) Global full self-attention. (b) Attention after splitting the input into small patches.

a d -dimension vector. Different from [9] and [10], before the input data are passed into the Transformer blocks, we do not need the convolutional layers to extract the feature map and increase the hidden size of the input sequence. This part of essential operation is replaced by individual neuron networks that project the patches. We also add position embedding to the projection to maintain position information in the data. The output of this projection is referred as patch embeddings.

B. Window Shifting

The patchwise self-attention splits the input series of samples into nonoverlapping patches. However, this approach breaks the data correlation at patch boundaries and ignores the connection across patches, which limit its modeling power. In order to capture the connection across patches while still maintaining the computational efficiency of nonoverlapping patches, we apply the shifted window technique to broaden the receptive field, which is inspired by [41]. As illustrated in Fig. 1, we cyclically shift the input $\mathbf{I} \in \mathbb{R}^{W \times d_{\text{model}}}$ to the right for $W/(2k)$ positions (i.e., half of the patch size); the right-most half-patch of samples are moved to the left-most part of the window. The patches obtained from the cyclically shifted window of data are also fed into patch embeddings as well, to create a patchwise feature map as shown in Fig. 1.

C. Transformer Layers

The two feature maps created by patch embeddings are then fed into the Transformer layers. We equally split the H heads

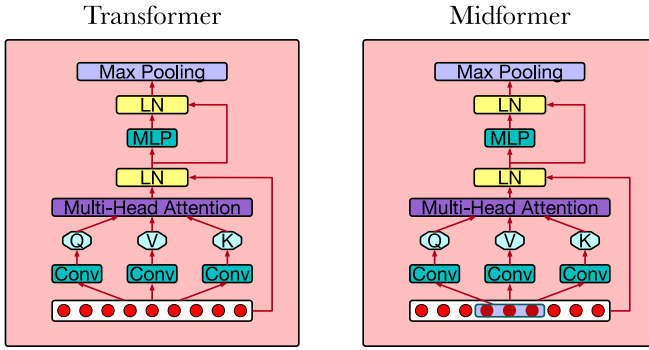


Fig. 3. Comparison of transformer and the proposed Midformer approach.

into two parallel groups, where each group has $H/2$ heads (assume that H is an even number). One group accepts the feature map created from the original input, and the other group accepts the feature map created from the shifted input.

We follow the Transformer layer designed in [6], which consists of a multihead attention layer and multilayer perception (MLP) layer. A LayerNorm (LN) layer is applied after each attention layer and the MLP layer. A residual connection is used from the input to the first LN layer, and from the first LN layer to the second LN layer. The architecture of the original Transformer model is shown in the left plot in Fig. 3.

We further enhance the existing Transformer layer and propose the Midformer layer, to achieve reduced computation complexity. The idea is simple: we only apply global attention on N (e.g., $N = 3$) input patches in the middle range of the window as queries, which is illustrated in the right plot in Fig. 3. The reason why we reduce the number of queries is that, most NILM models (e.g., s2p [12]) only predict the appliance’s power usage at the center position of the window. The middle range area of the window is where we should focus on. By reducing the number of queries, the complexity of the attention mechanism can be greatly reduced. It is worth noting that only the number of queries is reduced here, and the number of key–value pairs remains unchanged, which is fundamentally different from simply using a smaller window size W and then calculating the full attention. This technique contributes to the class of position-based sparse attention schemes, which reduce the required computations by limiting the number of query–key pairs that each query attends to [42].

D. Concatenation

Finally, an MLP (i.e., a fully connected layer) is utilized to concatenate the outputs of the two groups of Transformer blocks. The final MLP would restore the concatenated feature maps to the desired output size, which is one for NILM problems.

E. Computational Complexity Analysis

Supposing each input’s dimension is $W \times d_{\text{model}}$, the patch size is W/k , the learnable parameter’s dimension is $d_{\text{model}} \times d$, and the number of queries used in the Midformer layer is N . The computational complexity of the global Multihead Self-attention module in each layer is $\mathcal{O}(W^2 \cdot d)$. The high cost of

computing the global limits its ability to handle the usually large window sizes in NILM problems. However, with the proposed Midformer model, the computational complexity of the multihead self-attention module is reduced to $\mathcal{O}(N/k \cdot W \cdot d)$. In the Midformer design, both N and k are set proportional to the window size W (e.g., $k = W/9$ and $N = k/3 = W/27$ in our experiments). Therefore, the computational complexity of Midformer is now linear to the window size W .

V. EXPERIMENTAL STUDY

In this section, we introduce the data sets and the system configuration used in our experiments to evaluate the performance of the proposed Transformer model. We then present our experimental study of the proposed model and compare it with three baseline models.

A. Data Sets

We use two real-world data sets: 1) the REFIT data set [43] and 2) the UKDALE data set [44] to evaluate the performance of the proposed energy disaggregation method. The REFIT and UK-DALE data sets are both recorded in England. They both provide house-level aggregate energy consumption as well as individual appliances’ power consumption data. In particular, the REFIT data set consists of data from 20 households. Both the aggregate and appliance levels’ data were recorded every 8 s from September 2013 to July 2015. The UKDALE data set includes data from five houses. Each house’s aggregated energy consumption was recorded every 1 or 6 s, and the appliance level data was measured every 6 s. In order to be consistent with data in different data sets, the aggregate level and appliance level data are downsampled to 8 s. Standard score normalization is applied in data preprocessing; each sample x in the data set is normalized as $\hat{x} = (x - \bar{x})/S$, where \bar{x} is the sample mean and S is the sample standard deviation. We follow the approach in [12] to set the sample mean and sample standard deviation values for each appliance.

Following the approach in our recent work [2], for pretraining, we use a large-scale NILM data set: i.e., the REFIT data set. Specifically, we use the data from three houses as the pretraining set and the data from two other houses as the testing set for each appliance. The specific houses used and the amount of data from REFIT used to pretrain the model are summarized in Table I. We then use the UKDALE data set to evaluate the generalization of the models. We use only a small part of the data in House 2 of the UKDALE data set to fine-tune the pretrained model and the rest of the unseen data of House 2 to test the performance of the fine-tuned pretrained model. There is no overlap between the testing data and the fine-tuning data. The detailed information of the house and data from the UKDALE data set used in our experiment is summarized in Table II.

B. Model and Experimental Setup

Next, we introduce the experimental setup and the models used to address the NILM problem. The following three baseline models are evaluated for comparison purpose.

TABLE I
APPLIANCES AND HOUSES USED IN THE REFIT DATA SET [43]

Appliances	Training and validation dataset		
	House	Time period	Samples (M)
Kettle	5, 7, 13	2013-09-26 to 2015-07-08	18.91
Dishwasher	4, 10, 12	2014-03-07 to 2015-07-08	19.36
Fridge	2, 5, 12	2013-09-17 to 2015-07-08	13.28
Washing Machine	5, 7, 18	2013-09-17 to 2015-07-08	19.80
Microwave	5, 7, 18	2013-09-26 to 2015-07-08	19.80
Appliances	Testing dataset		
	House	Time period	Samples (M)
Kettle	9	2013-12-17 to 2015-07-08	6.17
	20	2014-03-20 to 2015-06-23	5.17
Dishwasher	9	2013-12-17 to 2015-07-08	6.17
	16	2014-03-10 to 2015-07-08	5.72
Fridge	9	2013-12-17 to 2015-07-08	6.17
	15	2013-12-17 to 2015-07-08	6.22
Washing Machine	15	2013-12-17 to 2015-07-08	6.22
	17	2014-03-06 to 2015-06-19	5.43
Microwave	17	2014-03-06 to 2015-06-19	5.43
	19	2014-03-06 to 2015-06-20	5.62

TABLE II
APPLIANCES AND HOUSES USED IN THE UKDALE DATA SET [44]

Appliances	Fine-tuning dataset		
	House	Time period	Samples (M)
Kettle, Dishwasher, Fridge, Washing Machine, Microwave	2	2013-5-20 to 2013-5-29	0.108
Appliances	Testing dataset		
	House	Time period	Samples (M)
Kettle, Dishwasher, Fridge, Washing Machine, Microwave	2	2013-5-30 to 2013-10-10	1.592

- 1) *s2p* [12]: This baseline model uses the same structure of the *s2p* method as in [12].
- 2) *Bidirectional Gated Recurrent Units (Bi-GRU)* [30]: This baseline model utilizes Bi-GRU, rather than LSTM, to reduce the amount of model parameters while maintaining a similar performance as the RNN model.
- 3) *Transformer (Transformer)* [6]: This is the traditional Transformer model. It has the same hyper-parameters as the Midformer model proposed in this article, which are summarized in Table III.

Note that comparisons between the *s2p* model and other traditional machine learning methods have been presented in [12], including AFHMM, RNN, sep2sep, GRU, etc., where the *s2p* model achieves the best performance. Therefore, we choose *s2p* as a benchmark scheme in this article.

All the models are implemented using TensorFlow 2.6.0 and trained on NVIDIA RTX 2070 Mobile. We pretrained all the models using the ADAM optimization algorithm [45] with a

TABLE III
HYPER-PARAMETER SETTING OF MIDFORMER

Hyper-parameter	Value
Window size	99 297 495 693
Batch size	100
Adam	0.001
Maximum pre-training epochs	50
Maximum fine-tuning epochs	10
Number of heads	8
Number of layers	2 to 4
Number of patches	3 9 11 99
Projected dimension	64

maximum of 50 gradient updates. We update the weights with a learning rate of 0.001 and use a minibatch size of 100. Both Midformer and Transformer incorporate 2–4 attention layers. The projected dimension of Midformer is $d = 64$, and the number of heads is $H = 8$. The number of patches is fixed at $k = 3, 9, 11, 99$. Table III describes the detailed information of the hyper-parameters.

We fine-tune the pretrained model using the stochastic gradient descent (SGD) method with a momentum of 0.9 and a learning rate of 0.01.

C. Performance Metrics

We use two metrics to evaluate the performance of the proposed Transformer model, which are the mean absolute error (MAE) and the signal aggregate error (SAE). These two metrics are defined as follows:

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^T |\hat{y}_j(t) - y_j(t)| \quad (4)$$

$$\text{SAE} = \frac{1}{r_j} |\hat{r}_j - r_j| \quad (5)$$

where T is the duration of the period used to predict the output; $y_j(t)$ is the ground truth of power consumption of appliance j and \hat{y}_j is the predicted value by the NILM models; and \hat{r}_j and r_j are the predicted total energy consumption and the ground truth of appliance j over period T , respectively.

D. Experimental Results and Discussions

Three scenarios are designed and examined in our experimental study, which are as follows

- 1) The pretrained model is evaluated on the same appliance in the same data set.
 - 2) The model is applied to a different data set but on the same appliance.
 - 3) The model learned using one appliance in one data set is evaluated on other appliances in a different data set.
- Multiple cases are examined, which belong to these three scenarios and use the data from the two public data sets.

1) *REFIT Data Set*: The results in terms of the evaluation metrics on the REFIT data set are represented in Table IV, which covers Scenario 1) described above. In this experiment, models for each appliance is pretrained using the REFIT training set. Next, the data for the same appliance from two unseen

TABLE IV
MODEL PERFORMANCES ON THE REFIT DATA SET

Kettle	Testing House 9		Testing House 20		Average	
	MAE	SAE	MAE	SAE	MAE	SAE
s2p	10.882	0.054	5.162	0.058	8.022	0.056
Bi-GRU	11.072	0.076	6.958	0.074	9.015	0.075
Transformer	7.874	0.052	3.971	0.056	5.923	0.054
Midformer	6.313	0.062	4.081	0.041	5.197	0.052

Dishwasher	Testing House 9		Testing House 16		Average	
	MAE	SAE	MAE	SAE	MAE	SAE
s2p	14.683	0.463	14.622	0.292	14.653	0.378
Bi-GRU	15.408	0.294	24.665	1.290	20.037	0.792
Transformer	14.159	0.463	21.533	1.348	17.846	0.906
Midformer	12.973	0.154	10.991	0.289	11.982	0.222

Fridge	Testing House 9		Testing House 15		Average	
	MAE	SAE	MAE	SAE	MAE	SAE
s2p	25.145	0.240	25.465	0.095	25.305	0.168
Bi-GRU	23.672	0.107	27.150	0.230	25.411	0.169
Transformer	29.153	0.366	23.970	0.507	26.562	0.437
Midformer	23.853	0.261	24.900	0.418	24.377	0.340

Washing Machine	Testing House 15		Testing House 17		Average	
	MAE	SAE	MAE	SAE	MAE	SAE
s2p	10.808	0.562	8.277	0.284	9.543	0.423
Bi-GRU	11.377	0.499	12.068	0.293	11.723	0.396
Transformer	10.573	0.562	10.373	0.244	10.473	0.403
Midformer	9.236	0.281	6.694	0.229	7.965	0.255

Microwave	Testing House 17		Testing House 19		Average	
	MAE	SAE	MAE	SAE	MAE	SAE
s2p	5.342	0.210	3.984	0.458	4.663	0.334
Bi-GRU	4.879	0.563	5.455	0.662	5.167	0.613
Transformer	6.505	0.594	4.507	0.266	5.506	0.430
Midformer	4.395	0.190	4.058	0.250	4.227	0.220

houses are used to evaluate the pretrained model. For example, for kettle, the labeled kettle data from Houses 5, 7, and 13 are used to pretrain the models, and then the kettle data from Houses 9 and 20 are used to test the pretrained models, while all the houses belong to the same REFIT data set. Table IV shows that the proposed Midformer model achieves both lower MAE and SAE in most cases (i.e., 6 cases out of 10 for MAE and 7 cases out of 10 for SAE). The average MAE and SAE values are averaged over the two houses. Our model achieved the best MAE results in all the cases, as well as the best SAE results for all the cases except for fridge. Compared to the baseline model s2p [12], the MAE reductions for kettle, dishwasher, washing machine, microwave, and fridge are 35.21%, 18.23%, 16.54%, 9.35%, and 3.80%, respectively.

Fig. 4 presents the execution times of different models for training per epoch under different window sizes. The Transformer and Midformer models both have two attention layers. The training set includes 100K samples. From the figure, we can see that the s2p model [12] uses the least amount of time; our proposed model uses the second least amount of time. The traditional Transformer model, which has the same number of layers as Midformer, consumes the longest time for training. The Bi-GRU model [30] uses less training time

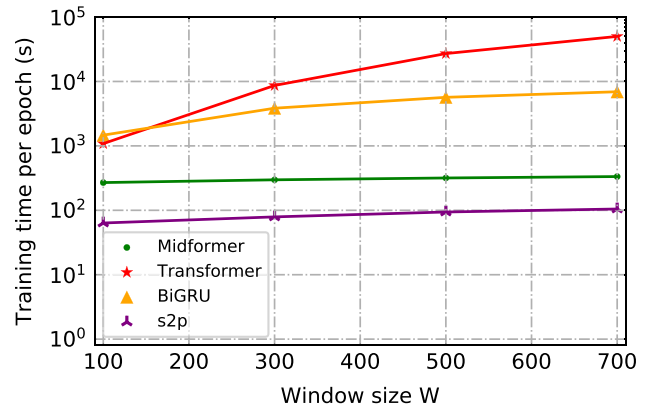


Fig. 4. Execution times of s2p [12], Bi-GRU [30], transformer (full attention) [6], and Midformer on the training set.

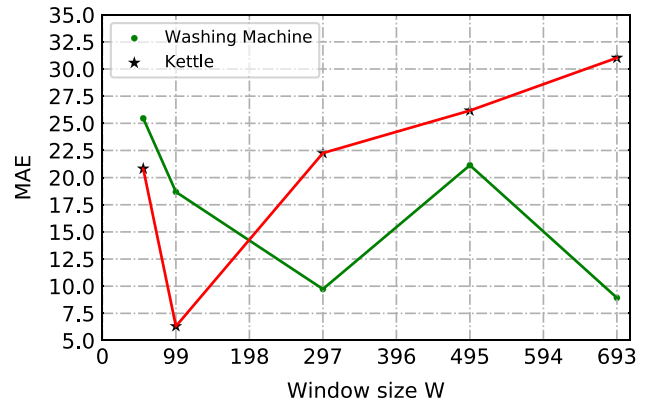


Fig. 5. MAEs obtained by Midformer models with different window sizes by testing the kettle in House 9 and the washing machine in House 15. The number of patches is set to 11.

TABLE V
BEST NUMBER OF PATCHES AND THE BEST PATCH SIZE UNDER DIFFERENT WINDOW SIZES (WASHING MACHINE)

Window size	99	297	693
Best number of patches	33	9	11
Best patch size	3	33	63
Best result (MAE)	10.88	8.93	8.55

TABLE VI
ABLATION STUDY: MAE RESULTS

Washing Machine	Midformer	Midformer w/o Window Shifting
House 15	8.55	9.12

than Transformer for most of the window sizes (except for $W = 100K$). However, it is still more time consuming, thank both s2p and Midformer. Considering the time and accuracy factors together, our proposed Midformer model consumes very little time for training and achieves the highest accuracy.

Next, we conduct an ablation study to further investigate the effectiveness of the proposed model. For brevity, we only present the results using the first week of the testing set. The window size and the number of patches are important parameters in our model structure. Fig. 5 presents a comparison of the MAEs obtained by Midformer models with different window

TABLE VII
RESULTS OF THE PRETRAINED MODEL WITHOUT FINE-TUNING TESTED ON THE UKDALE DATA SET

Pre-trained Dataset	Testing Dataset	Bi-GRU		s2p		Transformer		Midformer	
<i>REFIT</i>	<i>UKDALE</i>	MAE	SAE	MAE	SAE	MAE	SAE	MAE	SAE
Kettle	Kettle	16.579	0.163	22.642	0.442	16.204	0.268	15.099	0.234
	Dishwasher	72.909	0.498	66.627	0.669	70.220	0.562	71.449	0.558
	Fridge	47.079	0.783	44.102	0.786	46.458	0.798	46.625	0.795
	Washing machine	26.613	0.331	21.459	0.122	25.377	0.242	25.929	0.259
	Microwave	24.651	1.432	18.755	0.625	23.448	1.272	23.580	1.297
Dishwasher	Kettle	81.610	0.325	60.178	0.372	44.579	0.766	15.099	0.234
	Dishwasher	80.330	0.103	81.828	0.329	47.492	0.798	71.449	0.558
	Fridge	40.446	0.954	39.396	0.995	40.993	0.946	46.625	0.795
	Washing machine	44.862	0.601	41.812	0.939	44.541	0.632	45.929	0.259
	Microwave	44.487	0.294	44.362	0.362	42.864	0.263	43.580	1.297
Fridge	Kettle	371.049	8.047	318.502	6.556	306.362	6.264	303.152	6.127
	Dishwasher	379.803	3.338	305.632	4.484	308.999	4.471	313.630	4.569
	Fridge	25.034	0.321	25.621	0.873	24.273	0.218	25.702	0.222
	Washing machine	157.459	12.417	109.502	8.172	107.688	8.087	110.956	8.388
	Microwave	1224.569	28.545	169.378	20.964	168.284	20.870	172.590	21.404
Washing machine	Kettle	197.971	3.073	181.299	3.060	216.437	4.070	183.575	3.073
	Dishwasher	171.177	2.266	162.226	1.938	149.851	2.800	135.068	1.949
	Fridge	47.722	0.752	45.399	0.800	68.355	0.766	50.924	0.647
	Washing machine	31.097	1.337	19.867	0.432	58.687	3.966	33.079	1.359
	Microwave	79.898	8.825	66.123	7.040	99.997	11.766	65.980	7.091
Microwave	Kettle	113.396	1.038	119.416	1.233	56.051	0.424	116.196	1.096
	Dishwasher	121.416	0.515	129.421	0.670	62.974	0.624	129.026	0.627
	Fridge	39.345	1.000	41.747	0.919	39.344	1.000	41.670	0.929
	Washing machine	11.629	0.995	17.279	0.471	11.726	0.985	16.329	0.545
	Microwave	9.974	0.657	6.429	0.167	9.271	0.718	7.698	0.202
average		98.444	3.144	86.360	2.537	83.619	2.943	84.566	2.586

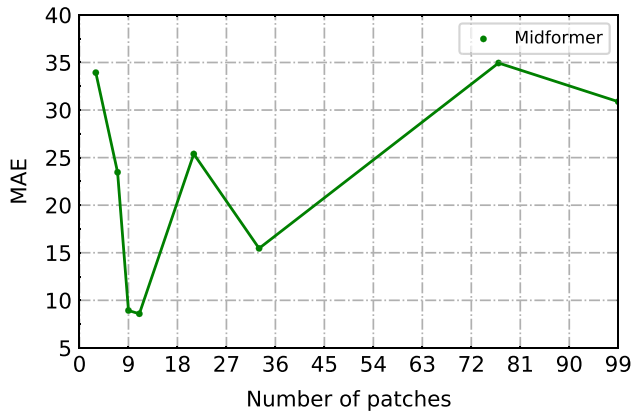


Fig. 6. MAEs achieved by Midformer models with different numbers of patches for washing machine, when the window size is set to 693.

sizes for the kettle in House 9 and the washing machine in House 15. The figure shows that the best window size for each appliance is different, which is 99 for kettle and 693 for washing machine. An overly large window size might hurt the disaggregation performance and increase the model's training time. Choosing a proper window size is vital for saving the training cost. Note that the unique windows size for different appliances limits the transferability of a trained model to different appliances. Therefore, during the fine-tuning process in the following part of the experiments, we adopt the same window size of the pretrained model for the fine-tuned model. We will explore the problem of transfer learning with different window sized models in our future work. Fig. 6 illustrates the effect of the number of patches on the model performance. We

use washing machine as the subject of this study and the window size is set to 693. The figure shows that the best number of patches for washing machine is 11. We further test the washing machine model with different window sizes and different patch numbers. Table V shows the best number of patches and the best patch size for each given window size (i.e., 99, 297, and 693), as well as the best MAE result. We find that a larger window size requires a larger patch size accordingly to achieve the best performance.

We next study the impact of window shifting on the Midformer performance. The results are given in Table VI, which is obtained for the washing machine in House 15 with a window size of 693 and a patch size of 63. We replace the cyclically shifted window with unshifted Transformer blocks, and find the performance drops by 6.67%.

2) *UKDALE Data Set*: In this experiment, we verify the transferability performance of the pretrained model across different domains (i.e., different data sets and appliances). We first fine-tune the pretrained model, which was originally learned using one appliance in the *REFIT* data set, with a small portion of new data from the *UKDALE* data set, and then use the test set of *UKDALE* to verify the performance of the model on the same or different appliance (see Table II). These experiments cover Scenarios 2) and 3) described above.

The performance of the pretrained models on the unseen *UKDALE* data set is presented in Tables VII and VIII. Table VII are the results of the pretrained models without fine-tuning, while Table VIII are the results of the pretrained models after fine-tuning, on the same appliance or an unseen appliance. The first column of the tables indicates

TABLE VIII
RESULTS OF PRETRAINED MODEL WITH FINE-TUNING TESTED ON THE UKDALE DATA SET

Pre-trained Dataset	Testing Dataset	Bi-GRU		s2p		Transformer		Midformer	
<i>REFIT</i>	<i>UKDALE</i>	MAE	SAE	MAE	SAE	MAE	SAE	MAE	SAE
Kettle	Kettle	10.064	0.116	9.854	0.106	10.585	0.318	4.183	0.041
	Dishwasher	39.001	0.409	4.463	0.017	14.511	0.153	5.135	0.040
	Fridge	34.475	0.025	24.296	0.015	36.476	0.139	17.081	0.061
	Washing machine	9.540	0.715	12.604	0.194	19.102	0.126	6.979	0.185
	Microwave	14.192	1.014	5.800	0.169	4.275	0.018	4.183	0.013
Dishwasher	Kettle	55.162	0.241	4.471	0.017	6.874	0.058	† 3.837	0.010
	Dishwasher	29.180	0.389	4.820	0.009	7.970	0.086	6.974	0.012
	Fridge	34.949	0.030	28.004	0.196	36.237	0.113	15.312	0.162
	Washing machine	10.137	0.758	11.812	0.328	9.490	0.328	5.301	0.049
	Microwave	18.649	0.516	3.762	0.187	5.414	0.128	3.177	0.065
Fridge	Kettle	64.616	0.440	8.009	0.023	5.159	0.018	4.520	0.043
	Dishwasher	36.583	0.173	5.639	0.031	6.094	0.014	5.251	0.006
	Fridge	24.059	0.032	13.798	0.083	16.588	0.144	† 13.132	0.050
	Washing machine	13.342	0.503	8.553	0.523	7.008	0.277	5.110	0.045
	Microwave	13.889	0.455	5.402	0.250	5.497	0.175	3.126	0.171
Washing machine	Kettle	37.098	0.435	5.994	0.027	7.482	0.014	5.213	0.003
	Dishwasher	30.852	0.266	† 4.254	0.012	5.377	0.003	5.406	0.040
	Fridge	35.310	0.110	24.393	0.285	34.425	0.322	14.992	0.032
	Washing machine	15.987	0.177	8.089	0.390	9.523	0.055	† 4.887	0.114
	Microwave	23.553	1.177	4.267	0.020	7.303	0.053	3.165	0.041
Microwave	Kettle	8.424	0.078	8.268	0.068	7.443	0.064	6.114	0.020
	Dishwasher	6.406	0.082	6.389	0.065	6.014	0.029	4.310	0.039
	Fridge	27.256	0.120	17.572	0.012	17.428	0.036	22.551	0.022
	Washing machine	14.767	0.661	13.904	0.380	6.806	0.374	5.456	0.059
	Microwave	5.456	0.126	5.289	0.090	4.541	0.039	† 2.630	0.028
Average		24.518	0.358	10.088	0.136	12.145	0.123	7.121	0.056

the appliance and data set learned by the pretrained model. The second column indicates the unseen test data set and corresponding appliances (same or different). The remaining columns are the MAEs and SAEs achieved by the four models.

From Table VII, we can see that the results of the pretrained models without fine-tuning have relatively large errors. When the pretrained model uses the same appliance as the test appliance, the test results are better than that using a different appliance. Except for the Bi-GRU [30] model, the other three models achieve similar MAE and SAE values, which are around 85 and 3, respectively.

From Table VIII, it is obvious that fine-tuning has been very effective in reducing the error of all the models on unseen data set and appliances, since both the MAEs and SAEs of all the models are greatly improved. For example, the average MAE of Midformer is reduced from 84.566 to 7.121, and the average SAE is reduced from 2.586 to 0.056, after fine-tuning (huge improvements). In the table, the bold numbers in each row indicate the *best result* among the four models obtained for the test set when using a pretrained model of a particular appliance. For example, for pretrained model using kettle in REFIT and the target appliance kettle in UKDALE, the Midformer model achieves the smallest MAE of 4.183 and the smallest SAE of 0.041. The number marked by symbol “†” indicates the *best model* for that target appliance among all the pretrained models. For example, for the target appliance kettle, the pretrained model of Midformer learned from

the source appliance dishwasher achieves the best MAE of 3.837. To better present the results, we have summarized such information in Table IX.

We can make the following observations from these results.

- 1) The proposed Midformer model outperforms all other models on average and in most specific cases.
- 2) Our proposed model achieves superior transferability performance, which means we can use the pretrained Midformer model using one appliance for all other target appliances, resulting greatly reduced cost for model pretraining.
- 3) In most cases, the best result for a target appliance is obtained with the model pretrained using the same appliance.

The best pretrained model for fridge, washing machine, and microwave in the UKDALE data set is the model learned from the same appliance in the REFIT data set, respectively. This is intuitive since the pretrained model will perform well if the test data and training data share similar features. In Table IX, the proposed Midformer model accounts for four of the five best results of transfer learning.

The predicted power consumption values of House 2 in the UKDALE data set for the five appliances obtained by the four pretrained models on the REFIT data set (i.e., s2p, Bi-GRU, Transformer, and Midformer) for a specific time period are plotted in Fig. 7, along with the corresponding ground truth values. Note that the “Aggregate” values are the input to these

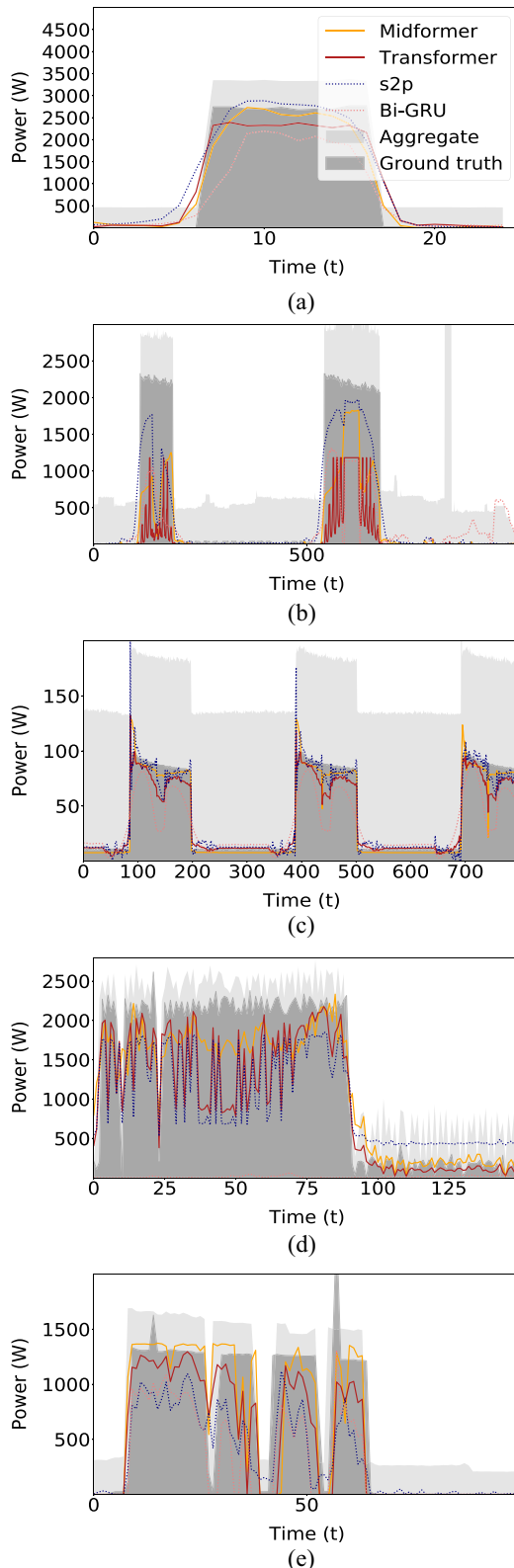


Fig. 7. Comparison of predicted power consumption values by Midformer, Transformer, s2p, and Bi-GRU for the five appliances, along with the ground truth values. (a) Kettle. (b) Dishwasher. (c) Fridge. (d) Washing machine. (e) Microwave.

models to be disaggregated into individual appliance's power consumption. The figure shows that the proposed Midformer model achieves the best performance compared to the three

TABLE IX
BEST PRETRAINED MODEL FOR THE UKDALE TEST SET

Test appliance in UKDALE	Pre-trained appliance in REFIT	Model	MAE	SAE
Kettle	Dishwasher	Midformer	3.837	0.010
Dishwasher	Washing machine	s2p	4.254	0.012
Fridge	Fridge	Midformer	13.132	0.050
Washing machine	Washing machine	Midformer	4.887	0.114
Microwave	Microwave	Midformer	2.630	0.028

baseline models, except for the dishwasher (which is consistent with the results in Table IX). The Bi-GRU model fails to predict the washing machine's power state at some time instances, i.e., the washing machine's state is on, but it is predicted as off [see Fig. 7(d)].

VI. CONCLUSION

In this article, we proposed the Midformer model to tackle the NILM problem. We utilized patchwise attention and reduced the query size to reduce the quadratic time complexity in traditional Transformer models to linear complexity. We also focused on the transferability performance of the models, which helped to reduce the model training cost and eased the deployment of the model in various environments. Our experimental study using two real-world data sets demonstrated the superior performance and stronger transferability of the proposed Midformer model over three baseline, state-of-the-art models on addressing the NILM problem.

REFERENCES

- [1] W. Li, T. Logenthiran, V.-T. Phan, and W. L. Woo, "Implemented IoT-based self-learning home management system (SHMS) for Singapore," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2212–2219, Jun. 2018.
- [2] L. Wang, S. Mao, B. M. Wilamowski, and R. M. Nelms, "Pre-trained models for non-intrusive appliance load monitoring," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 56–68, Mar. 2022, doi: [10.1109/TGCN.2021.3087702](https://doi.org/10.1109/TGCN.2021.3087702).
- [3] H. Zou, S. Mao, Y. Wang, F. Zhang, X. Chen, and L. Cheng, "A survey of energy management in interconnected multi-microgrids," *IEEE Access*, vol. 7, pp. 72158–72169, 2019.
- [4] Y. Wang, Y. Shen, S. Mao, X. Chen, and H. Zou, "LASSO & LSTM integrated temporal model for short-term solar intensity forecasting," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2933–2944, Apr. 2019.
- [5] A. Ruano, A. Hernandez, J. Ureña, M. Ruano, and J. Garcia, "NILM techniques for intelligent home energy management and ambient assisted living: A review," *MDPI Energies*, vol. 12, no. 11, p. 2203, Jun. 2019.
- [6] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 6000–6010.
- [7] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," Jun. 2021, *arXiv:2010.11929*.
- [8] S. Duan, W. Yang, X. Wang, S. Mao, and Y. Zhang, "Temperature forecasting for stored grain: A deep spatiotemporal attention approach," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 17147–17160, Dec. 2021.
- [9] Z. Yue, C. R. Witzig, D. Jorde, and H.-A. Jacobsen, "BERT4NILM: A bidirectional transformer model for non-intrusive load monitoring," in *Proc. 5th Int. Workshop Non-Intrusive Load Monitor.*, New York, NY, USA, Nov. 2020, pp. 89–93.
- [10] N. Lin, B. Zhou, G. Yang, and S. Ma, "Multi-head attention networks for nonintrusive load monitoring," in *Proc. IEEE Int. Conf. Signal Process. Commun. Comput.*, Macau, China, Aug. 2020, pp. 1–5.
- [11] C. Wu, F. Wu, T. Qi, and Y. Huang, "Fastformer: Additive attention can be all you need," Sep. 2021, *arXiv:2108.09084*.
- [12] M. D'Incecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1419–1429, Mar. 2020.

- [13] S. M. Tabatabaei, S. Dick, and W. Xu, "Toward non-intrusive load monitoring via multi-label classification," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 26–40, Jan. 2017.
- [14] P. Huber, A. Calatroni, A. Rumsch, and A. Paice, "Review on deep neural networks applied to low-frequency NILM," *MDPI Energies*, vol. 14, no. 9, p. 2390, Apr. 2021.
- [15] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," in *Proc. Int. Conf. Artif. Intell. Stat.*, La Palma, Spain, Apr. 2012, pp. 1472–1482.
- [16] M. Zhong, N. Goddard, and C. Sutton, "Signal aggregate constraints in additive factorial HMMs, with application to energy disaggregation," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 3590–3598.
- [17] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proc. AAAI*, Toronto, ON, Canada, Jul. 2012, pp. 356–362.
- [18] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. SIAM Int. Conf. Data Min.*, Mesa, AZ, USA, Apr. 2011, pp. 747–758.
- [19] B. Zhao, L. Stankovic, and V. Stankovic, "On a training-less solution for non-intrusive appliance load monitoring using graph signal processing," *IEEE Access*, vol. 4, pp. 1784–1799, 2016.
- [20] K. He, L. Stankovic, J. Liao, and V. Stankovic, "Non-intrusive load disaggregation using graph signal processing," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1739–1747, Aug. 2016.
- [21] G.-Y. Lin, S.-C. Lee, J. Y.-J. Hsu, and W.-R. Jih, "Applying power meters for appliance recognition on the electric panel," in *Proc. IEEE Conf. Ind. Electron. Appl.*, Taichung, Taiwan, Jun. 2010, pp. 2254–2259.
- [22] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Non-intrusive appliance load monitoring using low-resolution smart meter data," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Venice, Italy, Nov. 2011, pp. 31–40.
- [23] M. B. Figueiredo, A. De Almeida, and B. Ribeiro, "An experimental study on electrical signature identification of non-intrusive load monitoring (NILM) systems," in *Proc. Int. Conf. Adapt. Natural Comput. Algorithms*, Ljubljana, Slovenia, Apr. 2011, pp. 31–40.
- [24] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proc. AAAI*, New Orleans, LA, USA, Feb. 2018, pp. 1–8.
- [25] C. Shin, S. Joo, J. Yim, H. Lee, T. Moon, and W. Rhee, "Subtask gated networks for non-intrusive load monitoring," in *Proc. AAAI*, vol. 33, Honolulu, HI, USA, Jan. 2019, pp. 1150–1157.
- [26] K. Chen, Q. Wang, Z. He, K. Chen, J. Hu, and J. He, "Convolutional sequence to sequence non-intrusive load monitoring," *J. Eng.*, vol. 2018, no. 17, pp. 1860–1864, Nov. 2018.
- [27] K. Chen, Y. Zhang, Q. Wang, J. Hu, H. Fan, and J. He, "Scale- and context-aware convolutional non-intrusive load monitoring," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2362–2373, May 2020.
- [28] L. Mauch and B. Yang, "A new approach for supervised power disaggregation by using a deep recurrent LSTM network," in *Proc. IEEE GlobalSIP*, Orlando, FL, USA, Dec. 2015, pp. 63–67.
- [29] J. Kim, T.-T.-H. Le, and H. Kim, "Nonintrusive load monitoring based on advanced deep learning and novel signature," *Hindawi Comput. Intell. Neurosci.*, vol. 2017, Oct. 2017, Art. no. 4216281.
- [30] O. Krystalakos, C. Nalmpantis, and D. Vrakas, "Sliding window approach for online energy disaggregation using artificial neural networks," in *Proc. 10th Hellenic Conf. Artif. Intell.*, Patras, Greece, Jul. 2018, pp. 1–6.
- [31] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context aware energy disaggregation using adaptive bidirectional LSTM models," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3054–3067, Jul. 2020.
- [32] J. Kelly and W. Knottenbelt, "Neural NILM: Deep neural networks applied to energy disaggregation," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy-Efficient Built Environ.*, Seoul, South Korea, Nov. 2015, pp. 55–64.
- [33] R. Bonfigli, A. Felicetti, E. Principi, M. Fagiani, S. Squartini, and F. Piazza, "Denosing autoencoders for non-intrusive load monitoring: Improvements and comparative evaluation," *Elsevier Energy Build.*, vol. 158, pp. 1461–1474, Jan. 2018.
- [34] D. Murray, L. Stankovic, V. Stankovic, S. Lulic, and S. Sladojevic, "Transferability of neural network approaches for low-rate energy disaggregation," in *Proc. IEEE ICASSP*, Brighton, U.K., May 2019, pp. 8330–8334.
- [35] A. M. A. Ahmed, Y. Zhang, and F. Eliassen, "Generative adversarial networks and transfer learning for non-intrusive load monitoring in smart grids," in *Proc. IEEE SmartGridComm*, Tempe, AZ, USA, Nov. 2020, pp. 1–7.
- [36] Y. Liu, L. Zhong, J. Qiu, J. Lu, and W. Wang, "Unsupervised domain adaptation for nonintrusive load monitoring via adversarial and joint adaptation network," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 266–277, Jan. 2022.
- [37] I. J. Goodfellow, "On distinguishability criteria for estimating generative models," May 2015, *arXiv:1412.6515*.
- [38] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, Dec. 2016, pp. 2234–2242.
- [39] S. Wang, B. Z. Li, M. Khabba, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," Jun. 2020, *arXiv:2006.04768*.
- [40] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," May 2016, *arXiv:1409.0473*.
- [41] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," Aug. 2021, *arXiv:2103.14030*.
- [42] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," Jun. 2011, *arXiv:2106.04554*.
- [43] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study," *Sci. Data*, vol. 4, no. 1, pp. 1–12, Jan. 2017.
- [44] J. Kelly and W. Knottenbelt, "The U.K.-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five U.K. homes," *Sci. Data*, vol. 2, no. 1, pp. 1–14, Mar. 2015.
- [45] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," Jan. 2017, *arXiv:1412.6980*.