

# Adversarial Energy Disaggregation

ZHEKAI DU and JINGJING LI, School of Computer Science and Engineering, University of Electronic Science and Technology of China, China

LEI ZHU, School of Information Science and Engineering, Shandong Normal University, China

KE LU and HENG TAO SHEN, School of Computer Science and Engineering, University of Electronic Science and Technology of China, China

Energy disaggregation, also known as non-intrusive load monitoring (NILM), challenges the problem of separating the whole-home electricity usage into appliance-specific individual consumptions, which is a typical application of data analysis. NILM aims to help households understand how the energy is used and consequently tell them how to effectively manage the energy, thus allowing energy efficiency, which is considered as one of the twin pillars of sustainable energy policy (i.e., energy efficiency and renewable energy). Although NILM is unidentifiable, it is widely believed that the NILM problem can be addressed by data science. Most of the existing approaches address the energy disaggregation problem by conventional techniques such as **sparse coding**, **non-negative matrix factorization**, and the **hidden Markov model**. Recent advances reveal that deep neural networks (DNNs) can get favorable performance for NILM since DNNs can inherently learn the discriminative signatures of the different appliances. In this article, we propose a novel method named **adversarial energy disaggregation** based on DNNs. We introduce the idea of adversarial learning into NILM, which is new for the energy disaggregation task. Our method trains a generator and multiple discriminators via an adversarial fashion. The proposed method not only learns shared representations for different appliances but captures the specific multimode structures of each appliance. Extensive experiments on real-world datasets verify that our method can achieve new state-of-the-art performance.

CCS Concepts: • **Applied computing** → *Industry and manufacturing*; • **Information systems** → **Data mining**; **Data streaming**;

Additional Key Words and Phrases: Energy disaggregation, non-intrusive load monitoring (NILM), signal processing, data analysis

## ACM Reference format:

Zhekai Du, Jingjing Li, Lei Zhu, Ke Lu, and Heng Tao Shen. 2021. Adversarial Energy Disaggregation. *ACM/IMS Trans. Data Sci.* 2, 4, Article 27 (November 2021), 16 pages.  
<https://doi.org/10.1145/3477301>

This work was supported in part by the National Natural Science Foundation of China under grant 61806039 and 62073059, and in part by Sichuan Science and Technology Program under grant 2020YFG0080 and 2020YFG0481.

Authors' addresses: Z. Du, J. Li (corresponding author), K. Lu, and H. T. Shen, School of Computer Science and Engineering, University of Electronic Science and Technology of China, 2006 Xiyuan Ave., High-tech Zone West, Chengdu, Sichuan, China, 611731; emails: dzk1996411@163.com, lijn117@yeah.net, kel@uestc.edu.cn; L. Zhu, School of Information Science and Engineering, Shandong Normal University, No.1, University Road, Science Park, Changqing, Jinan, Shandong, China, 250358; email: leizhu0608@gmail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2021 Association for Computing Machinery.

2577-3224/2021/11-ART27 \$15.00

<https://doi.org/10.1145/3477301>

## 1 INTRODUCTION

Energy efficiency has been recognized as one of the twin pillars of sustainable energy [33], which is a great challenge facing humanity in the 21st century [34]. Today, most of the energy consumption behaviors have digital records and many energy problems can be formulated as informatics problems [22]. Thus, there is a growing expectation in our community that data science can play a role in addressing the energy challenge. In this article, we focus on energy disaggregation, a.k.a. **non-intrusive load monitoring (NILM)** [14], which is proven to have significant effects on energy efficiency [8, 31]. Specifically, energy disaggregation investigates the problem of separating an aggregated energy signal into the individual consumptions of different appliances. For a better understanding, we give an example of the NILM task in Figure 1. Previous studies [8, 11, 31] reported that estimating the appliance-level energy consumption data is able to reduce the energy consumption by as much as 15%.

Technically, energy disaggregation can be formulated as a single-channel **blind source separation (BSS)** problem [25, 39]. This problem is not trivial since it is unidentifiable (i.e., one needs to discover more than one source from a single observation). It was originally introduced by Hart in the 1990s [14] and has been an active research topic since then. As the demand of energy management has increased continuously, fruitful line of works have studied to address the NILM problem over the decades. A majority of existing methods either leverage the idea of signal processing that explicitly resorts to the features of appliances [6, 9, 15, 38], or using machine learning methods in supervised [18, 36, 40] and unsupervised manners [23, 41]. Specifically, many machine learning methods model the energy consumption of appliances by unsupervised probabilistic approaches such as **factorial hidden Markov models (FHMMs)** [41] and its variants [23], and other methods also deploy machine learning techniques such as sparse coding [22], matrix factorization [4], and  $k$ -nearest neighbor ( $k$ -NN) [37] to separate the energy signal. Most of these methods are event-driving (i.e., they track the footprints of different appliances to estimate whether a device is turned on or off). Recently, with the releases of many large-scale public datasets [19, 24], **deep neural networks (DNNs)** have been successfully applied to energy disaggregation in a supervised way. These methods are more like model-driving (i.e., they try to automatically reveal the temporal structure embedded in the observation data). For instance, Neural NILM [18] introduces **sequence-to-sequence (seq2seq)** learning into energy disaggregation and achieves remarkable performance improvement against traditional methods.

Inspired by the success of DNNs in energy disaggregation, we propose a novel method based on DNNs in this article. Specifically, we introduce adversarial learning into energy disaggregation. The idea of adversarial learning was reported in **generative adversarial networks (GANs)** [13]. GANs jointly train a generator and a discriminator via an adversarial manner. The generator synthesizes samples from noises to fool the discriminator. At the same time, the discriminator distinguishes whether a sample is real or fake. Once the discriminator is confused, it is considered that the generated samples have the same distribution as the real ones. The basic idea of our **adversarial energy disaggregation (AED)** is similar to GANs. Suppose that there is an expert (discriminator) who can recognize a specific electric appliance by its signal, and once the signal features learned by our model (generator) can fool the expert, it is safe to say that the learned features are effective. However, the original GANs cannot directly handle the energy disaggregation problem. At first, energy disaggregation is not a generative problem. It is a BSS problem. Second, in original GANs, the discriminator is served as a binary classifier, assigning fake or real to an input sample. However, in energy disaggregation, the discriminator should determine the source of a signal (i.e., corresponding appliance), which corresponds to a multi-class classification. To address the preceding issues, we formulated AED from the following two aspects. First, the generator is formulated as a feature representation network. Since different appliances share the

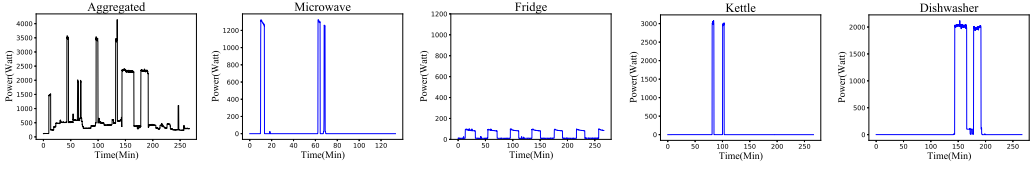


Fig. 1. An illustration of the task in this article. The sub-figures (from left to right) show the aggregated signal and the signals of the microwave, fridge, kettle, and dishwasher, respectively. The task of energy disaggregation is to decompose the aggregated signal into different appliance-specific signals.

same observation from the mains readings, we deploy a shared **convolutional neural network (CNN)** to learn the temporal features of appliances. Second, considering the complex multimode signal structures of different appliances, we train **multiple discriminators to enable precise and fine-grained source separation**. As a result, we formulate our AED as a multi-adversarial learning model that addresses the BSS problem.

Based on this design, we now sketch how our method can be applied in real-world scenarios, where variant appliances and houses exist. First, our model is trained on the large-scale supervised datasets, which inherently reduces the over-fitting risk and better generalizes to unseen houses in the wild to a certain extent. However, like other data-driven algorithms, the training model on a specific target appliance inevitably results in a generator that is appliance-biased (i.e., the model learns features that benefit the current appliance while being ineffectual for other appliances). Our method alleviates this issue by a multi-adversarial learning paradigm, which aims to exploit the common information across multiple appliance-specific features encoded by multiple feature extractors of appliances available at hand, hoping the learned features are effective for other unseen appliances in the real world without further fine-tuning. It is worth noting that although training a deep model is computationally expensive, it does not need to be performed very often—that is, once the model is trained, we only need the aggregated data and a lightweight forwarding inference to obtain the disaggregated energy consumption for each appliance, which is much faster and more economic than intrusive load monitoring approaches such as deploying smart sensor devices (e.g., smart plugs) for each appliance. As pointed out by Kelly and Knottenbelt [18], who first suggested applying DNNs to energy disaggregation, the inference could be performed on a compact compute device within each house without GPUs, and the energy cost of DNN inference is also marginal in comparison to the saved energy. On the contrary, smart plugs are expensive to install and hard to maintain simultaneously. In a nutshell, the main contributions of this work can be summarized as follows:

- (1) We propose a novel NILM method named *adversarial energy disaggregation* (AED). Different from previous DNN methods that use conventional CNNs and **recurrent neural networks (RNNs)**, AED introduces adversarial learning into the energy disaggregation problems. Experiments verify that AED can significantly outperform previous state-of-the-art algorithms.
- (2) We report a new CNN structure (reported later in Figure 3) that is able to learn discriminative feature representations for different appliances. At the same time, we present the multi-adversarial BSS framework (reported later in Figure 2) which enables fine-grained single-channel BSS.
- (3) Energy efficiency is a great challenge facing humanity in the 21st century. Our method develops a practical solution for NILM that takes steps toward energy efficiency.

The rest of this article is organized as follows. Section 2 makes a review of previous work related to this article. Section 3 presents the notations and elaborates on our proposed framework. In

Section 4, we conduct extensive experiments on two widely used energy disaggregation datasets and show the superiority of our method compared to existing methods. Finally, we present a conclusion and discussion in Section 5.

## 2 RELATED WORK

### 2.1 Energy Disaggregation

Energy disaggregation, which is also known as *non-intrusive load monitoring* (NILM), was first introduced by Hart [14]. The task of energy disaggregation is to separate individual signals of different appliances from an overall observation. In previous work, a battery of methods were explored to address the energy disaggregation problem. According to Shin et al. [36], **the most popular model for NILM is the FHMM [12]**. For instance, Zhong et al. [41] proposed an additive FHMM with signal aggregate constraints. Shaloudegi et al. [35] developed a scalable approximate inference FHMM algorithm based on a semi-definite relaxation combined with randomized rounding. Apart from FHMM, other techniques such as sparse coding [22] and support vector machines (SVMs) were also leveraged in the community [10]. Recently, DNNs have been introduced into energy disaggregation [5, 18, 36, 40]. Due to the superior performance in time series processing, **RNNs and their variants such as long short-term memory networks (LSTMs) and gated recurrent units (GRUs) have been employed in NILM**. For instance, Mauch and Yang [29] advocate using multiple bidirectional LSTM layers to tackle the problem of energy disaggregation. Kim et al. [20] address the energy disaggregation by an LSTM model as well as a novel signature to boost the performance. Kaselimi et al. [16] propose CoBiLSTM, which employs the representational power of LSTM networks and realizes the adaptation to the external environment.

Although the majority of studies that model time series data leverage RNNs, CNNs have also been adapted to the context of energy disaggregation due to its powerful ability in extracting local features of patterns. To alleviate the computational complexity issue in long input sequences, a common trick is to use a sliding window instead of the whole sequence. For instance, Kelly et al. [18] investigated several DNNs, such as CNNs, RNNs, and denoising autoencoders, to handle the energy disaggregation problem, they propose the seq2seq learning with slide windows and show the superior of deep learning methods to traditional methods. Chen et al. [7] proposed a convolutional seq2seq model and introduce the gated linear unit (GLU) convolutional blocks into energy disaggregation, which are used to extract information from the mains readings and control the features output by conventional CNN layers. Zhang et al. [40] further proposed a **sequence-to-point (seq2point)** network that **only predicts the middle point of the window**, which significantly improves the performance of previous SOTA methods.

### 2.2 Deep Adversarial Networks

The most popular deep adversarial learning paradigm is GANs [13], which consist of two neural networks known as the generator and discriminator, respectively. In the training process, the generator synthesizes fake samples to fool the discriminator. At the same time, the discriminator tries its best to distinguish the fake from the real.

It is worth noting that some studies have been carried out with the GANs in the literature. As an early attempt to leverage GANs in energy disaggregation, Bao et al. [1] proposed to integrate the generator of a pre-trained GAN into the conventional Neural NILM process to generate the appliance load sequences more accurately. Later, Kaselimi et al. [17] used a CNN-based seeder and generator to encode the mains signals and produce the appliance consumption, respectively, and with adversarial learning, the produced load sequences would match the ground truth as close as possible. Recently, Pan et al. [32] proposed a **sequence-to-subsequence (seq2subseq)** learning

in NILM, which makes a trade-off between seq2seq and seq2point learning and predicts a subsequence of the main window. To this end, conditional GAN [30] is used to encourage the generator to produce the appliance load sequences conditioned on the input mains windows.

In our model, although we share the similar spirit of GANs and tailor a model for energy disaggregation, our proposed method is significantly different from previous work in both motivation and formulation. Specifically, we leverage the idea of adversarial learning and report new contributions to both the generator and discriminator. On the one hand, we train multiple discriminators, one for each appliance, to capture the complex multimode structures. On the other hand, we propose a new deep network structure to learn the feature representations of different appliances. Furthermore, previous studies all use GAN for generating the whole appliance load sequences—that is, they employ an end-to-end learning model and generate the final predictions directly, and the adversarial process is carried out on the final output space. However, our model can be decomposed into a feature extractor (generator) that learns a latent feature space that is effective for energy disaggregation and a predictor (linear layer) that predicts the energy consumption based on the learned latent features, and the adversarial learning is carried out on the latent features output by the feature extractor. Thus, the reported work is significantly different from our AED.

### 3 ADVERSARIAL ENERGY DISAGGREGATION

In this section, we present our proposed method in detail. For a better understanding, we first review and formulate the problem of energy disaggregation. We present some notations and definitions that are used in our method. After that, we introduce the detail framework of our proposed AED.

#### 3.1 Problem and Formulation Overview

The goal of energy disaggregation is to recover the energy consumption of individual appliances from the mains power readings, which measure the whole-home energy consumption. Recent advances in energy disaggregation [18, 40] reveal that different appliances can be distinguished from the mains readings by learning deep features. Since energy disaggregation is an unidentifiable BSS problem, the signals of different appliances are mixed together in the mains readings. In the test process, we only have the aggregated signals for feature learning. Thus, the main challenge of feature-learning-based energy disaggregation methods is whether the learned features can capture the multimode structures of different appliances. For convenience, we refer to the features learned from the aggregated signals and appliance-specific signals as shared features and specific features, respectively. The specific features can identify the corresponding appliances. In our model, we encourage the shared features to capture the characteristics of each appliance. To this end, we propose the multi-adversarial learning approach as illustrated in Figure 2. Specifically, we first train multiple appliance-specific feature generators, one for each appliance, to learn the appliance-specific features. During the adversarial learning, the shared generator learn features to confuse the discriminators. Once the discriminators are confused, it is assumed that the shared feature representations have captured all the multimode structures of different appliances. In addition, a predictor (classifier) is also trained on the shared features to leverage the supervised information. In the following sections, we report the details of our proposed method.

#### 3.2 Notations and Definitions

Suppose there are  $N$  appliances in a household, and we observed the mains readings  $Y$  that represents the aggregate power of all appliances in watts, where  $Y = (y_1, y_2, \dots, y_T)$  and  $Y(t)$  denotes the mains readings at time  $t$ . For the  $i$ -th appliance, its power consumption is represented by a sequence  $X_i = (x_{i1}, x_{i2}, \dots, x_{iT})$  and  $X_i(t)$  denotes the assumption of the  $i$ -th appliance at time



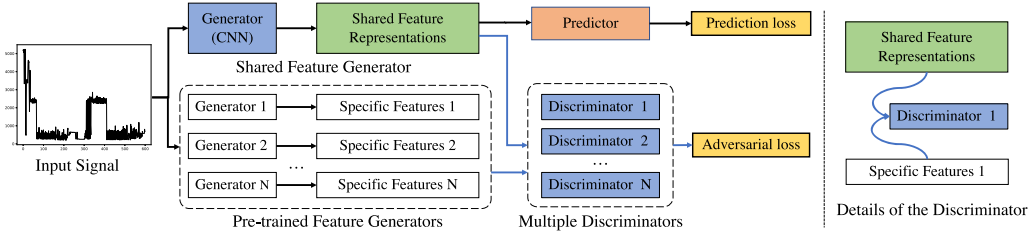


Fig. 2. Idea illustration. The left sub-figure shows the main idea of our method. The main architecture consists of a feature generator, a predictor, and  $N$  appliance discriminators, where  $N$  is the total number of appliances. During training, the appliance **discriminator distinguishes the shared features from the appliance-specific features**. Once a discriminator is confused, we assume that the shared features capture the latent structures of the corresponding specific features. For conciseness, the left figure is simplified. The right sub-figure shows the inputs of a discriminator. Here we take Discriminator 1 as an example. The rest can be done in the same manner. The appliance-specific feature generators, which have the same network structure with the shared generator, are pre-trained. More details about the networks are reported in the context.

$t$ . The relationship between  $X$  and  $Y$  can be represented by  $Y(t) = \sum_{i=1}^N X_i(t) + \epsilon(t)$ , where  $\epsilon(t)$  is the random noise that follows a Gaussian with mean 0 and variance  $\sigma_t^2$  (i.e.,  $\epsilon(t) \sim \mathcal{N}(0, \sigma_t^2)$ ). The task of energy disaggregation is to infer the individual consumptions of each appliance (i.e.,  $X_i = \{X_i(t)\}_{t=1}^T$ ) according to the mains readings  $Y$ . In our proposed DNNs, we use the letters  $G$ ,  $D$ , and  $C$  to denote the generator, the discriminator, and the predictor, respectively.

### 3.3 The Generators for Feature Representation

The feature learning process in deep energy disaggregation can be formulated as a seq2seq [18] or seq2point [40] learning problem. Theoretical analysis and experimental evaluation show that seq2point has better performance [40]. Specifically, seq2point trains a neural network to predict the midpoint power of an appliance when giving a window of the mains readings as the input. In this article, we follow the paradigm of seq2point learning. The feature learning network  $G$  (the generator of our adversarial model, parameterized by  $\theta_G$ ) takes a mains window  $Y_{t:t+W-1}$  as input and **outputs the midpoint power  $x_\tau$  of the corresponding window  $X_{t:t+W-1}$  of the target appliance**, where  $W$  is the window size and  $\tau = t + \lfloor W/2 \rfloor$ . The mapping can be described as  $x_\tau = G(Y_{t:t+W-1}) + \epsilon$ , and the loss function of this problem can be formulated as follows:

$$\mathcal{L}_G = \sum_{t=1}^{T-W+1} \log p(x_\tau | Y_{t:t+W-1}, \theta_G), \quad (1)$$

where  $\theta_G$  denotes the parameters of network  $G$ . This formulation is effective because it can make full use of the state information of the mains readings before and after the midpoint time to predict the power of the target appliance at the specific time. Theoretically, the problem in (1) can be maximized by many architectures, such as denoising autoencoders, CNNs, and RNNs. In this article, we propose a new CNN structure for energy disaggregation as shown in Figure 3. The proposed feature learning network consists of four convolutional layers and two max pooling layers. It is worth noting that the signal at time  $t$  is generally similar to the signal at  $t - \epsilon$  and  $t + \epsilon$ , where  $\epsilon$  is a small number. Therefore, the max pooling layers in our CNN can be used to avoid over-fitting and improve the generalization ability.

As illustrated in Figure 2, our model pre-trains multiple appliance-specific generators, one for each appliance, to learn the appliance-specific features. To this end, we also train a predictor  $C$

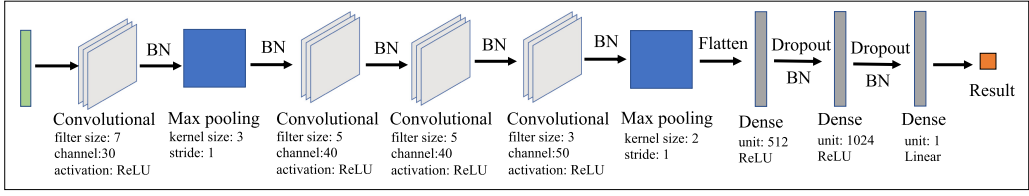


Fig. 3. An illustration of the structure and implementation details of the feature generator (the four convolutional layers and two max pooling layers) and the predictor (the three dense layers).

(parameterized by  $\theta_C$ ) that consists of three fully connected layers to leverage the supervised ground truth information. For a given appliance  $X_i$  and its ground truth information at time  $t$  (i.e.,  $x_{it}$ ), we deploy the following loss to train the appliance-specific generator  $G_i$  and the predictor  $C_i$  via an end-to-end manner:

$$\begin{aligned} & \mathcal{L}_{\text{pred}}(X_i, Y; G_i, C_i) \\ &= \mathbb{E}_{(x_{it}, y_{t-W/2:t+W/2}) \sim (X_i, Y)} \|x_{it} - C_i(G_i(y_{t-W/2:t+W/2}))\|_2^2. \end{aligned} \quad (2)$$

It is worth noting that the appliance-specific generators  $G_i$  ( $i \in [1, N]$ ) share the same structure, as shown in Figure 3, with the shared generator  $G$ . During the adversarial learning process, the pre-trained generators will be fixed to extract appliance-specific features.

### 3.4 Adversarial Energy Disaggregation

In the test stage, we only have the aggregated signal that consists of unidentified appliances—that is, we did not know which appliance contributes to the overall mains readings. Thus, it is necessary to learn a shared feature generator that is able to capture the multimode structure of individual signals.

To obtain a more generalized feature space, we try to find some latent common information from the load features encoded by multiple extractors of each appliance. Technically, we propose to use multiple-adversarial learning as illustrated in Figure 2. During the training process, we first use the  $N$  pre-trained extractors to extract  $N$  features for each mains readings window, then we train a generator  $G$  to compete against all the discriminators simultaneously. As the adversarial process continues, the generator will gradually learn to extract shared features and finally fool all the  $N$  discriminators [13]. Consequently, the feature representations learned in this way will be able to capture the multimode structure that embedded in the multiple appliance-specific feature spaces extracted by  $N$  pre-trained generators. The multi-adversarial learning process can be written as follows:

$$\begin{aligned} & \mathcal{L}_{\text{adv}}(Y; G, D_1, D_2, \dots, D_N) \\ &= \sum_{i=1}^N (\mathbb{E}_{y_{t-W/2:t+W/2} \sim Y} [\log(D_i(G(y_{t-W/2:t+W/2}))) \\ & \quad + \log(1 - D_i(G_i(y_{t-W/2:t+W/2})))]), \end{aligned} \quad (3)$$

where  $D_i$  represents the  $i$ -th discriminator that aims to distinguish whether the features come from shared generator  $G$  or the  $i$ -th pre-trained extractor  $G_i$ . After several rounds of training, the generator  $G$  would be able to extract shared feature representations for all the appliances. Figuratively speaking, features learned by the shared generator can be seen as “fake” to fool the discriminator, whereas features learned by the appliance-specific generators can be regarded as “real.” Once the discriminators are confused, the shared generator captures the complex multimode structures of each appliance.

**ALGORITHM 1:** Adversarial Energy Disaggregation

**Require:** Sequence  $X$  of the target appliance, mains readings sequence  $Y$ ,  $N$  pre-trained appliance-specific generators  $G_1, G_2, \dots, G_N$ ,  $N$  discriminators  $D_1, D_2, \dots, D_N$ , trainable target generator  $G$  and predictor  $C$ , hyper-parameter  $\lambda_{cls}$ , batch size  $B$ , maximum epoch  $M$ .

**Ensure:** Optimal  $\theta_G, \theta_C$  for the target model  $G$  and  $C$ .

```

1: for epoch  $\leftarrow 1$  to  $M$  do
2:   for batch  $\leftarrow 1$  to  $K$  do
3:      $g_G \leftarrow 0$ 
4:     Sample  $B$  main windows  $\{y_i\}_{i=1}^B \sim Y$  and obtain the corresponding  $x_i$  of each  $y_i$ ;
5:     Extract the features for  $\{y_i\}_{i=1}^B$  via  $\{G(y_i)\}_{i=1}^B$ ;
6:     for  $j \leftarrow 1$  to  $N$  do
7:       Extract the appliance-specific features for  $\{y_i\}_{i=1}^B$  via  $\{G_j(y_i)\}_{i=1}^B$ 
8:       Update  $D_j$  via  $\theta_{D_j} \leftarrow \text{Adam}(\nabla_{\theta_{D_j}}(\mathcal{L}_{adv}))$  according to Equation (3).
9:        $g_G += \nabla_{\theta_G}(-\mathcal{L}_{adv})$ 
10:    end for
11:    Calculate  $\mathcal{L}_{pred}$  according to Equation (2).
12:    Update  $G$  via  $\theta_G \leftarrow \text{Adam}(g_G + \nabla_{\theta_G}(\lambda \mathcal{L}_{pred}))$ 
13:    Update  $C$  via  $\theta_C \leftarrow \text{Adam}(\nabla_{\theta_C}(\lambda \mathcal{L}_{pred}))$ 
14:  end for
15: end for

```

At last, since the adversarial model is required to decompose the mains rather than only learning feature representations, we further train a predictor (classifier) in the adversarial learning framework. As a result, we have the overall formulation:

$$\min_{G,C} \max_{D_1, D_2, \dots, D_N} \mathcal{L}_{AED} = \mathcal{L}_{adv} + \lambda \mathcal{L}_{pred}, \quad (4)$$

where  $C$  denotes the classifier and  $\lambda > 0$  denotes the trade-off parameter. It is easy to see that the whole training process of our proposed AED is via an adversarial fashion. On the one hand, the generator  $G$  and classifier  $C$  are trained together to minimize prediction loss and multi-adversarial domain generalization loss. On the other hand, the domain discriminators  $D_1, D_2, \dots, D_N$  are trained together to maximize the multi-adversarial domain generalization loss so as to compete with the generator.

## 4 EXPERIMENTS

In this section, we verify the proposed method on two real-world datasets collected from U.S. and UK families. We compare AED with several previous state-of-the-art approaches that deploy different techniques. Our model is implemented by PyTorch and trained on NVIDIA GTX 2080Ti GPUs. The datasets used in this article can be downloaded via the provided links reported in Section 4.1. Our code and data are released at <https://github.com/lijin118/>.

### 4.1 Datasets

We testify our AED on two popular datasets for energy disaggregation. The description is listed as follows.

The REDD [24] dataset<sup>1</sup> is a widely used benchmark for NILM tasks. The dataset records the domestic energy consumption, at both appliance-level and whole-house level, of six U.S. houses

<sup>1</sup><http://redd.csail.mit.edu/>.



Table 1. Parameters Used in This Article for Processing the Data

Appliance	Window Length	Mean	Standard Deviation
Aggregate	599	522	814
Kettle	599	700	1,000
Microwave	599	500	800
Fridge	599	200	400
Dishwasher	599	700	1,000
Washing machine	599	400	700

The length unit is point and the power unit is watt.

from November 2012 to January 2015. The recording intervals of the appliance and mains readings are 3 seconds and 1 second, respectively. Following previous work [18, 40], we use houses 2 through 6 for training and house 1 for test. For the same reasons as those in previous work [40], we only consider the microwave, fridge, dishwasher, and washing machine in this paper.

The *UK-DALE* [19] dataset<sup>2</sup> records both appliance-level and whole-house level energy consumption of five UK houses from November 2012 to January 2015. The records are read every 6 seconds. In this article, we follow the previous work [40] and choose the washing machine, kettle, microwave, dishwasher, and fridge for evaluations. Houses 1, 3, 4, and 5 are used for training, and house 2 is used for test.

## 4.2 Implementation Details

*Network architecture.* Our model consists of three main components: the generator  $G$ , the discriminator  $D$ , and the predictor  $C$ . The implementation details of  $G$  and  $C$  are reported in Figure 3. The generator  $G$  consists of four convolutional layers and two max pooling layers. Specifically, these two max pooling layers are set after the first and the last convolutional layers and have pool sizes of 3 and 2, respectively. The filter size and channel of the convolutional layers are set to  $\{7 \times 1, 5 \times 1, 5 \times 1, 3 \times 1\}$  and  $\{30, 40, 40, 50\}$ , respectively. A replication pad is used to make the sequence length invariant after convolutional operation. The discriminator  $D$  is implemented by three fully connected (FC) layers (i.e., FC-ReLU-FC-ReLU-FC-Sigmoid). We adopt batch normalization and dropout in this work. The pre-trained  $G_i$  share the same structure with  $G$ . Different discriminators also have the same network architecture.

*Hyper-parameter setting.* We optimize our networks by the Adam [21] optimizer, and the parameters are  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 10^{-8}$ . The learning rate is 0.001. The window size is 599 (a sample window contains 599 recording points) and the batch size is 1,000. We set the number of maximum epoch to 50. The hyper-parameter  $\lambda = 0.05$ .

*Data pre-processing.* For fair comparisons, we follow previous work [40] to pre-process data. Specifically, we first align the mains readings with the appliance readings by timestamps in each house then concatenate them together. After that, we perform a normalization on the raw data by

$$\frac{x_t^i - \bar{x}^i}{\sigma^i}, \quad (5)$$

where  $x_t^i$  denotes the power reading at time  $t$  of the  $i$ -th appliance.  $\bar{x}^i$  and  $\sigma^i$  stand for the mean value and the standard deviation of the  $i$ -th appliance, respectively. The specific parameters for normalization can be found in Table 1. The normalized data is then fed into the model for training.

<sup>2</sup><https://jack-kelly.com/data/>.

### 4.3 Compared Methods and Evaluation Metrics

We compare our method with several previous representative approaches that deploy different techniques. Specifically, AFHMM [23] is a traditional method that is built on additive FHMMs. Seq2seq [18, 40] is a deep method that leverages seq2seq learning. Seq2point [40] is a recently reported deep methods that deploys seq2point learning. Seq2subseq [32] is a trade-off method between seq2seq and seq2point, which employs conditional GAN to learn the output distribution for each appliance. We cite the results of the batch normalization version (i.e., N-I-U) from Pan et al. [32] for a fair comparison. **Subtask gated networks (SGNs)** [36] explicitly consider the on/off states of appliances and combine two subtask networks (i.e., a regression network and a classification network) to make final predictions. In addition, we also compare our method with two benchmark algorithms implemented in the publicly available NILM toolkit (NILMTK) [2] platform, namely **combinatorial optimization (CO)** and FHMM. The experiments of CO and FHMM are conducted based on the rewritten experiment API of NILMTK provided by Batra et al. [3] under the same setting as other algorithms. It is worth noting that the reported results of seq2point are the best results we can achieve by running the authors' code. For fair comparisons, we follow previous work [40] and use the following two metrics:

$$\begin{aligned} \text{MAE} &= \frac{1}{T} \sum_{t=1}^T |\hat{x}_t - x_t|, \\ \text{SAE} &= \frac{|\hat{r} - r|}{r}, \end{aligned} \tag{6}$$

where  $\hat{x}$ ,  $x$ ,  $\hat{r}$ , and  $r$  denote the predicted consumption of an appliance at time  $t$ , the ground truth consumption of an appliance at time  $t$ , predicted total energy consumption of an appliance, and the ground truth total consumption of an appliance, respectively. It is easy to observe that  $r = \sum_t x_t$  and  $\hat{r} = \sum_t \hat{x}_t$ . It is worth noting that the used **mean aggregate error (MAE)** and **signal aggregate error (SAE)** have no linear relationship. The MAE reflects the fine-grained performance of prediction at every recording point. The normalized SAE reports a more global prediction accuracy.

### 4.4 Quantitative Results

The quantitative results on REDD and UK-DALE are reported in Table 2 and Table 3, respectively. From the results in Table 2, we can see that our method achieves the best results in two out of four appliances. In terms of average performance, our AED has both the smallest MAE and SAE. The average MAE of our AED is 17.68, whereas the previous baseline method seq2point achieves only 24.24. The improvement is 27%. We can also notice that our method significantly reduces the MAE of the dishwasher by 48.7% compared to seq2point that also uses the seq2point architecture. For other appliances like the microwave and washing machine, our method can reduce the MAE value by 39.5% and 28.5%, respectively. Furthermore, our method also decreases the average SAE by 44.4%. The results verify that our method has a much better NILM performance with respect to both timely prediction (MAE) and overall prediction (SAE).

The results in Table 3 draw similar conclusions. Our AED outperforms the previous state-of-the-art methods in four out of five appliances. Compared with seq2point, our AED reduces the average MAE and SAE by 43.3% and 64.0%, respectively. Specifically, the MAE values of the kettle, microwave, fridge, dishwasher, and washing machine are reduced by 34.8%, 31.6%, 36.2%, 49.3%, and 54.7%, respectively. Comparing with seq2subseq, our method shows significant improvements on the kettle, fridge, dishwasher, and washing machine, and achieves a comparable result on the microwave with respect to MAE. It is also worth noting that our method has the smallest standard deviations, which indicates that our method is robust for different appliances.

Table 2. Quantitative Results on the REDD Dataset

Metric	Methods	Microwv	Fridge	Dishwsh	Washmch	Average
MAE	AFHMM [23]	<b>11.85</b>	69.80	155.25	14.25	62.78 $\pm$ 58.20
	CO [2]	62.85	78.50	108.24	90.63	85.06 $\pm$ 37.25
	FHMM [2]	71.12	89.67	99.79	65.77	81.59 $\pm$ 42.14
	SGN [36]	17.52	<b>23.89</b>	14.97	20.07	19.09
	Seq2seq [40]	33.27	30.63	19.45	22.86	26.55 $\pm$ 5.61
	Seq2point [40]	29.602	34.118	22.476	16.130	24.24 $\pm$ 7.45
	AED (Ours)	17.914	29.770	<b>11.527</b>	<b>11.533</b>	<b>17.68 <math>\pm</math> 7.44</b>
SAE	AFHMM [23]	0.84	0.99	7.19	0.07	2.28 $\pm$ 2.86
	CO [2]	3.48	0.11	4.56	2.77	2.73 $\pm$ 1.53
	FHMM [2]	3.07	0.37	3.53	1.11	1.77 $\pm$ 0.87
	SGN [36]	—	—	—	—	—
	Seq2seq [40]	0.24	0.11	0.56	0.51	0.36 $\pm$ 0.183
	Seq2point [40]	<b>0.037</b>	0.100	0.701	0.240	0.27 $\pm$ 0.26
	AED (Ours)	0.171	<b>0.079</b>	<b>0.339</b>	<b>0.029</b>	<b>0.15 <math>\pm</math> 0.11</b>

Best results are highlighted in bold. For a better layout, we use *Microwv*, *Dishwsh*, *Washmch* to denote microwave oven, dishwasher, and washing machine.

Our method trains the feature representation network by multi-adversarial learning. Thus, the multimode structures embedded in different appliance-specific signals can be preserved, which explains the small standard deviation of our results. Later, we report the percentage of total energy consumption of different methods in Figure 6. The results give a straightforward report of the family energy usage in terms of different device types. We can see that our AED is able to generate a report that is very close to the ground truth.

#### 4.5 Qualitative Results

For a better understanding of the proposed method, we further report the qualitative results of our method and seq2point [40] in Figure 4 and Figure 5. Specifically, we visualize the signals of different observations in a recording period. Notably, we report the results of our AED without multi-adversarial learning, which is denoted as AED<sup>-</sup> in the figures, in the third column, which can be seen as a result of the ablation study. Comparing seq2point with our AED, it is easy to observe that our method can better fit the ground truth. From the results in the last columns and the third columns, we can see that multi-adversarial learning can further improve the performance of AED. Moreover, in the figures of the fridge, the seq2point shows severe fluctuations, whereas the curve obtained by our method is considerably smooth, indicating that our model is very stable and is able to filter the noise introduced by other appliances effectively.

In addition, we manually added two green boxes in the figures of the washing machine in Figure 5. For the sake of narration, we name the areas in the left green box and right green box as zone 1 and zone 2, respectively. It can be seen from the mains readings that zone 1 and zone 2 are similar in a way (e.g., similar consumption value and similar duration). However, the consumption in zone 1 is contributed by the washing machine (reflected by the ground truth line) and zone 2 is contributed by other appliances. It can be seen that our AED is able to distinguish such a subtle difference, whereas other methods cannot, which verifies the superiority of our AED formulation.

Table 3. Quantitative Results on the UK-DALE Dataset

Metric	Methods	Kettle	Microwv	Fridge	Dishwsh	Washmch	Average
MAE	AFHMM [23]	47.38	21.28	42.35	199.84	103.24	$89.79 \pm 64.50$
	CO [2]	47.65	100.26	78.76	109.96	91.52	$85.63 \pm 60.20$
	FHMM [2]	45.23	46.08	57.81	50.18	70.13	$54.09 \pm 36.33$
	SGN [36]	7.08	6.26	15.79	15.50	12.31	11.39
	Seq2subseq (N-I+U) [32]	10.34	<b>5.77</b>	35.44	28.68	20.65	20.18
	Seq2seq [18]	13.000	14.559	38.451	237.96	163.468	$93.49 \pm 91.11$
	Seq2point [40]	8.656	8.700	20.894	29.724	12.724	$16.14 \pm 8.12$
SAE	AED (Ours)	<b>5.645</b>	5.948	<b>13.324</b>	<b>15.062</b>	<b>5.764</b>	$9.15 \pm 5.72$
	AFHMM [23]	1.06	1.04	0.98	4.50	8.28	$3.17 \pm 2.88$
	CO [2]	1.60	11.81	1.22	1.83	2.04	$3.70 \pm 2.24$
	FHMM [2]	8.77	4.60	0.34	0.39	1.74	$3.17 \pm 2.02$
	SGN [36]	—	—	—	—	—	—
	Seq2subseq (N-I+U) [32]	0.081	0.365	0.477	0.709	0.317	0.390
	Seq2seq [18]	0.085	1.348	0.502	4.237	13.831	$4.00 \pm 5.12$
	Seq2point [40]	0.072	0.430	0.121	0.376	0.245	$0.25 \pm 0.14$
	AED (Ours)	<b>0.046</b>	<b>0.146</b>	<b>0.038</b>	<b>0.163</b>	<b>0.053</b>	$0.09 \pm 0.11$

Best results are highlighted in bold.

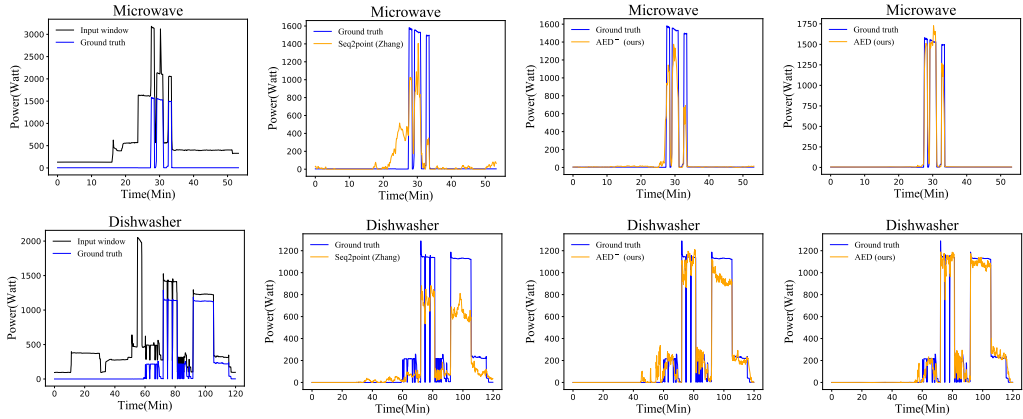


Fig. 4. Qualitative results on REDD. From left to right, the first column shows the results of mains readings and the ground truth of one specific appliance (e.g., dishwasher and microwave). The following columns report the results of seq2point, ours without multi-adversarial learning, and our full model.

#### 4.6 Model Discussion

*Ablation study.* Our method consists of the feature representation part and the multi-adversarial learning part. Figure 4 and Figure 5 report the results of our method without the multi-adversarial learning, denoted as AED<sup>+</sup>. In Table 4, we report the comprehensive ablation study of our model. It is obvious that multi-adversarial learning is beneficial to reduce MAE in all appliances and brings an average reduction on MAE by 1.76.

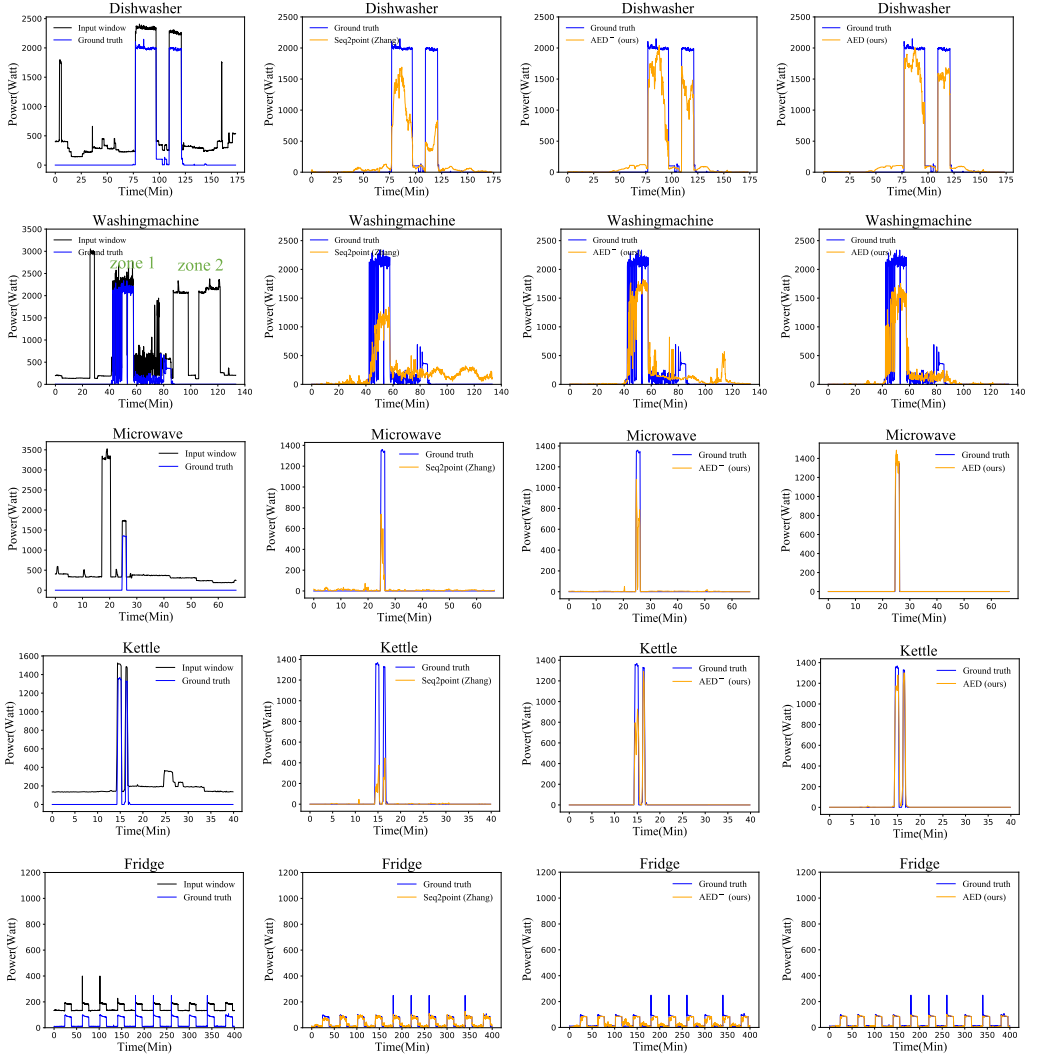


Fig. 5. Qualitative results on UK-DALE. From left to right, the first column shows the results of mains readings and the ground truth of one specific appliance. We chose all appliances in UK-DALE. The following columns report the results of seq2point, ours without multi-adversarial learning, and our full model.

*Training process.* To show the training process of AED, we report both the validation loss and the training loss of AED on the microwave of UK-DALE in Figure 7(a). As an adversarial model, it can be seen that our method is relatively stable during the training and it can achieve convergence in a few epochs.

*Parameter sensitivity.* Our method involves a hyper-parameter  $\lambda$  that is tuned on the validation set. Figure 7(b) reports the sensitivity of  $\lambda$  on the UK-DALE dataset. We tune the  $\lambda$  from 0.01 to 100. It can be seen that  $\lambda$  is not sensitive, and our model has a high tolerance for the variation of hyper-parameter.

Table 4. Ablation Study on REDD and UK-DALE Datasets in Terms of MAE

Dataset	Methods	Microwv	Fridge	Dishwsh	Washmch	Average
REDD	AED <sup>-</sup>	21.253	31.122	13.896	12.791	$19.76 \pm 7.32$
	AED (Ours)	17.914	29.770	11.527	11.533	$17.68 \pm 7.44$
UK-DALE	AED <sup>-</sup>	6.924	15.546	17.642	6.985	$11.78 \pm 6.09$
	AED (Ours)	5.948	13.324	15.062	5.764	$10.02 \pm 6.11$

We take the microwave and fridge of UK-DALE as examples.

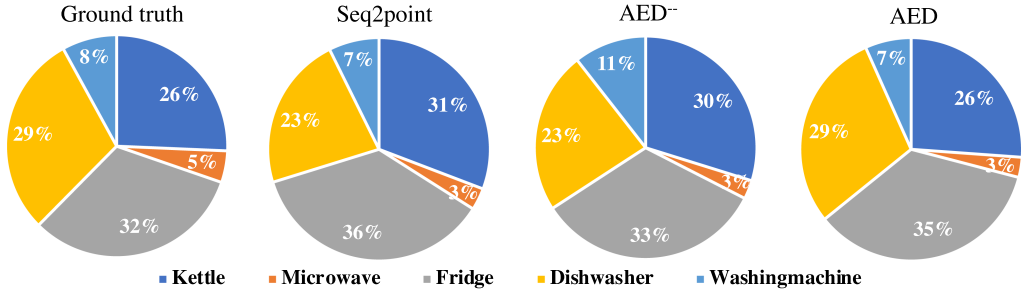


Fig. 6. Percentages of total energy usage in the test set of UK-DALE. From left to right, the four figures show the results of ground truth, seq2point, AED without multi-adversarial learning, and AED.

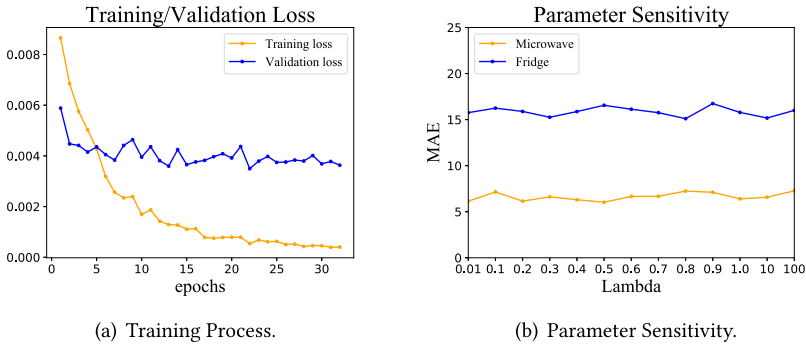


Fig. 7. Model analysis. (a) Training process of the proposed method. We take the microwave of UK-DALE as an example. (b) Parameter sensitivity of the proposed method.

## 5 CONCLUSION AND DISCUSSION

In this article, we propose a novel method named *adversarial energy disaggregation* (AED) for NILM. Specifically, we introduce the idea of adversarial learning into solving energy disaggregation problems. To learn better feature representations and capture the complex multi-mode structures of various appliances, we report a new CNN architecture and a multi-adversarial learning paradigm. Extensive experiments on two real-world datasets verify that our AED is able to achieve the new state-of-the-art results. In our proposed model, we need to pre-train several models (e.g., the discriminators) according to the number of appliances. Although it brings additional work, the total number of appliances is small for a typical family, and we can reuse



the pre-trained models for different families. For instance, we can use transfer learning [26–28] to challenge the problem. In our further work, we will explore transferring the pre-trained model across different families and try to recognize unseen appliances.

Last, the proposed method in this article is a non-intrusive monitoring method, which means that one can predict the electric appliances of a family without access into the house. For some reason, the electric appliances in a family could be private information. However, patterns of energy use can reflect the behavior patterns of human beings. For instance, the states of lights, such as turn on or off, could indicate whether there are people at home. Thus, it is suggested to use the technology under the condition of being fully acknowledged by the resident.

## REFERENCES

- [1] Kaibin Bao, Kanan Ibrahimov, Martin Wagner, and Hartmut Schmeck. 2018. Enhancing neural non-intrusive load monitoring with generative adversarial networks. *Energy Informatics* 1, 1 (2018), 295–302.
- [2] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. 2014. NILMTK: An open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th International Conference on Future Energy Systems*. 265–276.
- [3] Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odysseas Krystalakos, Mingjun Zhong, Paulo Meira, and Oliver Parson. 2019. Towards reproducible state-of-the-art energy disaggregation. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 193–202.
- [4] Nipun Batra, Hongning Wang, Amarjeet Singh, and Kamin Whitehouse. 2017. Matrix factorisation for scalable energy breakdown. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.
- [5] Gissella Bejarano, David DeFazio, and Arti Ramesh. 2019. Deep latent generative models for energy disaggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 850–857.
- [6] Mario Berges, Ethan Goldman, H. Scott Matthews, Lucio Soibelman, and Kyle Anderson. 2011. User-centered nonintrusive electricity load monitoring for residential buildings. *Journal of Computing in Civil Engineering* 25, 6 (2011), 471–480.
- [7] Kunjin Chen, Qin Wang, Ziyu He, Kunlong Chen, Jun Hu, and Jinliang He. 2018. Convolutional sequence to sequence non-intrusive load monitoring. *Journal of Engineering* 2018, 17 (2018), 1860–1864.
- [8] Sarah Darby. 2006. *The Effectiveness of Feedback on Energy Consumption: A Review for DEFRA of the Literature on Metering, Billing and Direct Displays*. Environmental Change Institute, University of Oxford.
- [9] Shengli Du, Mingchao Li, Shuai Han, Jonathan Shi, and Heng Li. 2019. Multi-pattern data mining and recognition of primary electric appliances from single non-intrusive load monitoring data. *Energies* 12, 6 (2019), 992.
- [10] Anthony Faustine, Nerey Henry Mvungi, Shubi Kaijage, and Kisangiri Michael. 2017. A survey on non-intrusive load monitoring methodologies and techniques for energy disaggregation problem. arXiv:1703.00785.
- [11] Corinna Fischer. 2008. Feedback on household electricity consumption: A tool for saving energy? *Energy Efficiency* 1, 1 (2008), 79–104.
- [12] Zoubin Ghahramani and Michael I. Jordan. 1996. Factorial hidden Markov models. In *Advances in Neural Information Processing Systems*. 472–478.
- [13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2672–2680.
- [14] George William Hart. 1992. Nonintrusive appliance load monitoring. *Proceedings of the IEEE* 80, 12 (1992), 1870–1891.
- [15] Yuanwei Jin, Eniye Tebekaemi, Mario Berges, and Lucio Soibelman. 2011. A time-frequency approach for event detection in non-intrusive load monitoring. In *Signal Processing, Sensor Fusion, and Target Recognition XX*, Vol. 8050. International Society for Optics and Photonics, 80501U.
- [16] Maria Kaselimi, Nikolaos Doulamis, Athanasios Voulodimos, Eftychios Protopapadakis, and Anastasios Doulamis. 2020. Context aware energy disaggregation using adaptive bidirectional LSTM models. *IEEE Transactions on Smart Grid* 11, 4 (2020), 3054–3067.
- [17] Maria Kaselimi, Athanasios Voulodimos, Eftychios Protopapadakis, Nikolaos Doulamis, and Anastasios Doulamis. 2020. EnerGAN: A generative adversarial network for energy disaggregation. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'20)*. IEEE, Los Alamitos, CA, 1578–1582.
- [18] Jack Kelly and William Knottenbelt. 2015. Neural NILM: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. 55–64.
- [19] Jack Kelly and William Knottenbelt. 2015. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data* 2, 1 (2015), 1–14.

- [20] Jihyun Kim, Thi-Thu-Huong Le, and Howon Kim. 2017. Nonintrusive load monitoring based on advanced deep learning and novel signature. *Computational Intelligence and Neuroscience* 2017 (2017), Article 4216281.
- [21] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.
- [22] J. Zico Kolter, Siddharth Batra, and Andrew Y. Ng. 2010. Energy disaggregation via discriminative sparse coding. In *Advances in Neural Information Processing Systems*. 1153–1161.
- [23] J. Zico Kolter and Tommi Jaakkola. 2012. Approximate inference in additive factorial HMMs with application to energy disaggregation. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*. 1472–1482.
- [24] J. Zico Kolter and Matthew J. Johnson. 2011. REDD: A public data set for energy disaggregation research. In *Proceedings of the Workshop on Data Mining Applications in Sustainability (SIGKDD'11)*. 59–62.
- [25] Te-Won Lee, Michael S. Lewicki, and Terrence J. Sejnowski. 2000. ICA mixture models for unsupervised classification of non-Gaussian classes and automatic context switching in blind signal separation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 10 (2000), 1078–1089.
- [26] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. 2020. Maximum density divergence for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Early access, April 28, 2020.
- [27] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, and Heng Tao Shen. 2019. Locality preserving joint transfer for domain adaptation. *IEEE Transactions on Image Processing* 28, 12 (2019), 6103–6115.
- [28] Jingjing Li, Ke Lu, Zi Huang, Lei Zhu, and Heng Tao Shen. 2018. Heterogeneous domain adaptation through progressive alignment. *IEEE Transactions on Neural Networks and Learning Systems* 30, 5 (2018), 1381–1391.
- [29] Lukas Mauch and Bin Yang. 2015. A new approach for supervised power disaggregation by using a deep recurrent LSTM network. In *Proceedings of the 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP'15)*. IEEE, Los Alamitos, CA, 63–67.
- [30] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. arXiv:1411.1784.
- [31] B. Neenan, J. Robinson, and R. N. Boisvert. 2009. *Residential Electricity Use Feedback: A Research Synthesis and Economic Framework*. Technical Report. Electric Power Research Institute.
- [32] Yungang Pan, Ke Liu, Zhaoyan Shen, Xiaojun Cai, and Zhiping Jia. 2020. Sequence-to-subsequence learning with conditional GAN for power disaggregation. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'20)*. IEEE, Los Alamitos, CA, 3202–3206.
- [33] Bill Prindle, Maggie Eldridge, Mike Eckhardt, and Alyssa Frederick. 2007. *The Twin Pillars of Sustainable Energy: Synergies between Energy Efficiency and Renewable Energy Technology and Policy*. American Council for an Energy-Efficient Economy, Washington, DC.
- [34] Cynthia Rosenzweig, David Karoly, Marta Vicarelli, Peter Neofotis, Qigang Wu, Gino Casassa, Annette Menzel, et al. 2008. Attributing physical and biological impacts to anthropogenic climate change. *Nature* 453, 7193 (2008), 353–357.
- [35] Kiarash Shaloudegi, András György, Csaba Szepesvári, and Wilsun Xu. 2016. SDP relaxation with randomized rounding for energy disaggregation. In *Advances in Neural Information Processing Systems*. 4978–4986.
- [36] Changho Shin, Sunghwan Joo, Jaeryun Yim, Hyoseop Lee, Taesup Moon, and Wonjong Rhee. 2019. Subtask gated networks for non-intrusive load monitoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1150–1157.
- [37] Seyed Mostafa Tabatabaei, Scott Dick, and Wilsun Xu. 2016. Toward non-intrusive load monitoring via multi-label classification. *IEEE Transactions on Smart Grid* 8, 1 (2016), 26–40.
- [38] A Longjun Wang, B. Xiaomin Chen, C. Gang Wang, and D. Hua. 2018. Non-intrusive load monitoring algorithm based on features of V-I trajectory. *Electric Power Systems Research* 157 (2018), 134–144.
- [39] Fa-Yu Wang, Chong-Yung Chi, Tsung-Han Chan, and Yue Wang. 2009. Nonnegative least-correlated component analysis for separation of dependent sources by volume maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 5 (2009), 875–888.
- [40] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. 2018. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.
- [41] Mingjun Zhong, Nigel Goddard, and Charles Sutton. 2014. Signal aggregate constraints in additive factorial HMMs, with application to energy disaggregation. In *Advances in Neural Information Processing Systems*. 3590–3598.

Received December 2020; revised April 2021; accepted July 2021