GraphNILM: A Graph Neural Network for Energy Disaggregation

Rui Shang[1], Siji Chen[1], Zhiqian Chen[2], and Chang-Tien Lu[1]

[1] Virginia Polytechnic Institute and State University Blacksburg, VA 24061-0002
[2] Mississippi State University, 75 B.S. Hood Drive, MS 39762
{srui,sijic,ctlu}@vt.edu
zchen@cse.msstate.edu

Abstract. Non-Intrusive Load Monitoring (NILM) remains a critical issue in both commercial and residential energy management, with a key challenge being the requirement for individual appliance-specific deep learning models. These models often disregard the interconnected nature of loads and usage patterns, stemming from diverse user behavior. To address this, we introduce GraphNILM, an innovative end-to-end model that leverages graph neural networks to deliver appliance-level energy usage analysis for an entire home. In its initial phase, GraphNILM employs Gaussian random variables to depict the graph edges, later enhancing prediction accuracy by substituting these edges with observations of appliance interrelationships, stripping the individual load enery from the aggregated main energy all at one time, resulting in reduced memory usage, especially with more than three loads involved, thus presenting a time and space-efficient solution for real-world implementation. Comprehensive testing on popular NILM datasets confirms that our model outperforms existing benchmarks in both accuracy and memory consumption, suggesting its considerable promise for future deployment in edge devices.

Keywords: NILM · Energy Disaggregation · Graph Neural Network

1 Introduction

Energy conservation is a crucial research area in today's scientific world. Household and commercial electrical use accounts for approximately 60% of global energy consumption [7]. Real-time monitoring of power consumption is a useful approach for assisting homes, utilities, appliance manufacturers, and policymakers in making more informed decisions. However, obtaining individual appliance-level load data in real-time typically requires the installation of a sensor per load, which can be costly and impractical for older houses or office buildings. As a result, non-intrusive load monitoring (NILM) technology has gained popularity due to its low installation and maintenance costs, as well as its respect for privacy. By gathering data from the main power measurements and computing the projected individual power consumption without additional measuring equipment, NILM provides a cost-effective solution for disaggregating energy consumption. Several surveys [1] have demonstrated the business case for NILM, revealing that energy savings outweigh installation costs. Moreover, research has shown that providing active energy data feedback to customers through NILM can reduce energy use by 5-20% [24]. However, NILM is inherently challenging due to the various load combinations in a given place and consumers' complex consumption preferences. Addressing this problem requires the development of innovative and efficient models that can accurately disaggregate energy consumption at the appliance level, which is the focus of this study. Figure 1 depicts the NILM application scenario utilizing state-of-the-art disaggregation techniques versus our proposed GraphNILM method. In most houses or office buildings, the number of routinely used devices exceeds four. As the number of appliances increases, the amount of resources needed to estimate the instantaneous load power rises. GraphNILM, in contrast, utilizes roughly the same amount of memory size to achieve comparable results, which seems more reasonable to be deployed on edge devices in the houses or office buildings.

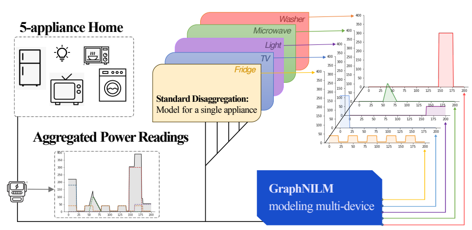

Fig. 1. NILM general explanations and comparisons of our proposed method with common state-of-art methods.

There are three main challenges in the NILM field. **(1) A low rate of sampling.** The sampling rate in the NILM field, which is typically 1 Hz for common datasets, is significantly lower than the working frequency of the loads; and thus, the sampled data cannot be fully restored according to the Nyquist-Shannon sampling theorem. Adoption of classical algorithms, like hidden Markov models and their variants [17, 22], yields restricted results under specific conditions, making widespread adoption challenging. **(2) Homogeneous data with restricted characteristics**. The majority of available data consists solely of aggregated power readings in a timely order, which makes it difficult for domain experts to quantify the dedicated load power numbers from readings only. Numerous studies therefore focus on the classification problem [8] by Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN) [5], rather than quantitative analysis, but even good classification results are of limited utility to consumers. Non-stable power consumption and the complex combinations of loads makes disaggregation a challenging problem. **(3) More memory resources required as more loads are introduced** In the NILM commercial sector, a regression model with instantaneous individual load power output appears more desirable than a categorization model. In recent years, nonlinear regression models in NILM have emerged with respectable performance, but the vast majority of deep learning models have a high memory footprint and increased computational complexity[23]. To improve performance, the majority of regression models [26, 13] repeat the same structure with different parameters for different loads. A residence with ten loads will necessitate the concurrent operation of ten times the proposed model, indicating the high cost of business implementations on edge devices. This research aims to develop an end-to-end model with reduced memory consumption and cutting-edge performance for future business deployments on edge devices. To solve the challenges highlighted in the NILM field, this paper designs the GraphNILM model and provides the following significant contributions:

- **Formulating a novel end-to-end framework for energy disaggregation.** The paper presents GraphNILM, a model that uses a modified convolutional neural network for initial disaggregation and a graph neural network for refinement, enabling simultaneous disaggregated power readings. GraphNILM efficiently extracts power features from low-rate sequences, fine-tuning the results using load relationships before producing the final output.
- **Constructing an effective algorithm to characaterize load relations.** This paper categorizes relationships between distinct loads as synchronous and asynchronous. Supplementing our approach, we introduce a new algorithm to calculate the synchronous relations between the aggregated power readings and individual load by correlations, and asynchronous relations between loads by dynamic matching.
- **Designing a new structure for memory reduction.** In response to single-load targeted models, we construct a weighted graph for GraphNILM, transforming loads into nodes using pre-established relationships. This allows simultaneous power disaggregation, reducing memory and computational requirements by avoiding separate individual load trainings. We are the first to use dynamic time wrapping relationship structure in the NILM field.
- **Conducting extensive experimental performance evaluations.** The proposed GraphNILM network has been evaluated utilizing data from standard NILM datasets: REDD [16] and UK-DALE [14]. It often surpasses competing methods across different metrics, utilizing a fraction of memory compared to benchmarks. We also discuss the practical benefits of integrating GraphNILM into edge devices.

2 Related Work

The NILM field was pioneered by Hart [10] about three decades ago. In order to solve the NILM problem with low rate sampling data, Hidden Markov based Model(HMM, FHMM, etc.) [17] and its variants[22] were adopted in the early stage. Such methods were categorized as event-based methods, which usually contain three procedures: edge detection, feature extraction , and classification [9, 18], and were broadened to handle NILM disaggregations problems in other fields [6]. With deep learning flourishing in most domains in recent years, deep neural networks and convolution neural networks have lowered the obstacles in the NILM field for researchers to extract power features without the help of domain experts [13]. Deep learning has lighted up a new direction for solving the NILM problem[19]. Long-short-term memory network (LSTM)[20] extracted dominate appliance usage from the aggregate power signals, which are collected at a low sampling rate. The widely-used Seq2Point model[26], showed great improvement on regression tasks in the NILM domain and received challenges all the time since its debut, but still need trainings per load introduced[4]. The introduction of graph signal processing (GSP) to the NILM field is a novel concept. Stankovic's research group [27, 11, 2] tracked this technique by segmenting aggregated energy sequences to do classification task in NILM. Similar work has been conducted by Bing and other groups [25], with all of them utilizing graph neural networks to perform load identification tasks which extracted information is insufficient to meet individual and commercial energy planning requirements.

Our aim is to create an easily implementable real-world model that can produce instantaneous disaggregated load energy with reduced memory consumption, while maintaining accuracy equivalent to the state-of-the-art techniques. Realized the ignorance of relations among loads in regression tasks and the difficulty of transforming homogeneity data to adapt to the multi-variate inputs needs for graph signal processing, we decide to design a new integrated framework for energy disaggregation by utilizing the advantages of the deep convolution neural network and the graph neural network. As a result, the proposed model can not only fine-tune the disaggregated power results, but also reduce total computations by incorporating the graph design.

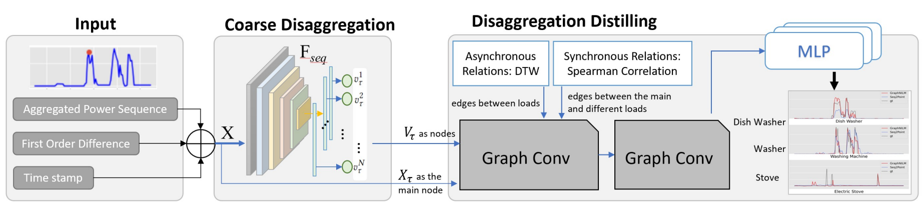

Fig. 2. GraphNILM Total Structure.

3 Proposed Model

3.1 Problem Setup: Disaggregation

Given the aggregated power meter readings, x_t, we wish to disaggregate the immediate power contribution of each individual load. The disaggregation of $x_t \in R$ at time t is formulated as:

$$x_t = \sum_{n=1}^{N} y_t^n + \text{noise}, \tag{1}$$

N is the number of loads which are to be monitored and y_t^n stands for the power of the n-th load at time t. The noise and unmonitored loads in the given dataset have been generalized to the niose. Our purpose is to get the individual load power together at the same time, as $Y_t = [y_t^1, y_t^2, ..., y_t^N]^T$ based on the measurement of the aggregated power x_t from our proposed model.

3.2 GraphNILM

The proposed GraphNILM model consists of two major components: coarse disaggregation and disaggregation distillation, see Fig 2. It utilizes convolutional neural networks to extract the representations of power from the aggregated power sequence. The output of the coarse disaggregation component represents the power characteristics of various loads and will be used as the coarse disaggregation results for the disaggregation distilling's inputs. GCN components in our disaggregation distillation section will use these inputs along with the constructed relations as the graph edges to disaggregate the main power to the individual load power in the given house.

(1) Coarse Disaggregation

The coarse disaggregation part is for extracting power characteristics. The input $x_{t-W+1:t}$ is a length W time series representing the aggregated main power within W time stamps. Then we intentionally add the first order difference between two consecutive aggregated main power readings which brings approximately half sigma better performance in the later experiment. F_{seq} represents the convolutional layers in the proposed architecture, and it outputs

$$V_\tau = F_{seq}(x_{t-W+1:t}) = [v_\tau^1, v_\tau^2, ..., v_\tau^N]^T, \tag{2}$$

where $\tau = t - \frac{W-1}{2}$ representing the middle point of the window W time series, and V_τ stands for the extracted power characteristics of the input sequence $x_{t-W+1:t}$. The intuition behind the midpoint selection is based on the assumption that the model can learn the information of aggregated power before and after the midpoint [21]. These results will be further used as the nodes in the graph structure in the following Disaggregation Distilling part.

(2) Disaggregation Distilling

The introduction of GraphNILM's second component, graph structure, is a novel concept in the NILM field, as the accessible data in popular datasets include no relational information. A graph $\mathcal{G}$ usually consists of nodes set $\mathcal{V}$ and adjacency matrix A and is represented as $\mathcal{G} = \{\mathcal{V}, A\}$. $v_i \in \mathcal{V}$ denotes for the i-th node in $\mathcal{V}$, which is v_τ^i from equation (2). The adjacency matrix defines the edges $a_{ij} \in A$ and their weights in the graph. These weighted edges are mapping our expectations to utilize relations among loads in the design of our proposed model. Some loads have simultaneous direct relationships, while others may have asynchronous relations. Weighted edges in a graph can appropriately describe such relationships when they can be quantified.

To leverage the relational strengths of the graph model, we map both the aggregated power and the coarse disaggregated characteristics - obtained from the coarse disaggregation stage - onto the nodes in $\mathcal{G}$, signifying that $x_\tau, V_\tau \subseteq \mathcal{V}$. To avoid over-fitting, we add x_τ, the mid-point of the aggregated power sequence, as the central node in the graph.

Finding meaningful edges between nodes, i.e., interpreting the relationships between loads in NILM, is the core principle in our graph construction. According to our observations, there are primarily two sorts of relationships in given houses: synchronous and asynchronous. For example, in a given house, the owner prefers to watch television with her food prepared. Before turning on the television, she toasts a slice of bread and then boils some water in the kettle. These events occur sequentially and have strong relationships from an asynchronous perspective: the individual power reading peak for one load occurs close to the power reading peaks for other loads. On the other hand, it is straightforward to conclude that the aggregated power x_τ closely connected with each load at the same time, e.g. the aggregated power would rise at the same time she turns on a new load, a typical synchronous relationship. Though for each house, owners' habits may vary, the major loads for the house are still similar, which makes our pre-trained model transferrable.

For *synchronous* relations, spearman correlations can be used in this model since it measures the strength and direction of monotonic association between two variables. Thus we use it to denote weight edges from the aggregated main power meter to the disaggregaged individual load. For *asynchronous* relationships, dynamic time wrapping (DTW) is used to determine the correlations between each load. In both of the typical datasets with which we explored, missing values at different time for different loads hampered our ability to obtain relationships through simple correlation techniques.Therefore, using DTW is a good choice for asynchronous relations. For the calculation of the distance between two load sequences, we define the k-th load with p samples as s_o^k and the l-th load with q samples as s_o^l: $s_o^k = [y_1^k, ..., y_p^k]$ and $s_o^l = [y_1^l, .., y_q^l]$. However, since the power ranges of each load are different and DTW accumulates the absolute distance, meaning two asynchronous well-correlated loads with small power may get smaller results than two asynchronous uncorrelated loads with large power. Therefore, we must normalize the load sequence in order to have meaningful DTW results:

$$s^k = \frac{s_o^k}{E(s_o^{k_{ON}})}, \tag{3}$$

where $E(s_o^{k_{ON}})$ is the mean of the k-th load's active power when the load turned ON. Then, using DTW algorithm [11], we will obtain the final DTW distance as $\mathcal{D}(s^k, s^l)$. Apply the same rule to all loads and we will get DTW standard distances between loads. Next, we translate standard distances, ranging from 0 to ∞ to range 0 to 1, to ensure the asynchronous and synchronous relations lie in the same range, by

$$r_d = e^{-\mathcal{D}(s^k, s^l)/\alpha}, \tag{4}$$

where α is a scale factor chosen manually based on the average distances. The relations between the load and the aggregated main power, r_s, and the relations between the load and the other load, r_d, may thus be applied to the weighted edges in A in our graph $\mathcal{G}$. The entire process of how the adjacency matrix A is derived from loads and main power is shown in the algorithm 1. The proposed method utilizes two weeks of known load data $S = [s^1, ..., s^N]$ concatenated from equation 3 in the given house to perform our algorithm. The standard distance between the k-the load and l-th load is the element positioned at the k-the row/column l-th column/row in A. By resampling the aggregated power sequence X to X^k, which has aligned samples with s^k in the given time, spearman correlation r_s^k can be performed simultaneously. Then the r_s^k could be placed in A's k-th row/column $N+1$-th column/row. With two relations being calculated and put into the appropriate locations in A, the building of the adjacency matrix for the designed weighted graph is completed.

By mapping our design into standard GCN layer[15], our proposed method completes the initial distilling. With the repeated GCN layer nodes fully connected to the MLP layer at the output, the GraphNILM will return N results representing the disaggregated power readings for N separate loads.

Algorithm 1: Adjacency matrix from DTW and correlation

input : X, S
output: The adjacency matrix A
Start $\alpha = 1000$, $A = [\mathbf{1}]_{N+1,N+1}$
while *not all edges,* r_s, r_d*, in* A *have been computed* **do**
    $\mathcal{D}(0,0) \leftarrow 0$ ▷ Initialize the start point
    s^k, $s^l \sim S$ ▷ Sample s^k, s^l from S
    $r_d = \exp(-\mathcal{D}(s^k, s^l)/\alpha)$ ▷ Equation (4)
    X^k, X^l ▷ Resample X to match s^k, s^l
    r_s^k, r_s^l ▷ X^k, s^k and X^l, s^l to perform correlation
    $A(k,l) \leftarrow r_d, \quad A(l,k) \leftarrow r_d$
    $A(N+1,k) \leftarrow r_s^k, \quad A(k,N+1) \leftarrow r_s^k$
    $A(N+1,l) \leftarrow r_s^l, \quad A(l,N+1) \leftarrow r_s^l$
end

4 Experiment

This study involves the examination of two mainstream open-access datasets: REDD [16], UK-DALE [14]. All datasets have labeled appliance-level power consumption along with whole-house power consumption. We also use NILMTK [3] for data prepossessing and comparing results among benchmarking algorithms. All these algorithms are implemented in Python3 and run on NVIDIA QUADRO P5000 GPU. The model is implemented using Pytorch and Pyg. To ensure the consistency of our results, each experiment was performed 20 times with a fixed seed. The Adam optimizer was employed, with a learning rate 0.005 and the batch size is 1024. The training and testing split is 0.8 vs 0.2. The loss function applied in the proposed model is L2 loss. Standard normalization is used on both input and output with data from each appliance normalized separately. Any negative values post-denormalization are set to 0. An early stop was implemented after 49 non-improved validation losses. The reason for setting the threshold at 49 for monitoring validation loss is due to fluctuations observed in the validation loss. Initially, the model fits into the mean value of each appliance, causing the loss to decrease. However, once the mean is fitted, the model begins the disaggregation process, thereby inducing an increase in validation loss. The validation loss subsequently decreases once the disaggregation pattern is discerned. In addition to Seq2Point and GraphNILM model, Seq2MultiPoint model whose structure is largely similar to that of GraphNILM except the GCN layers is also tested for ablation study. GraphNILM* is for investigating the proposed model without first-order difference. This paper also evaluates the classical FHMM method for comparison.

4.1 Dataset

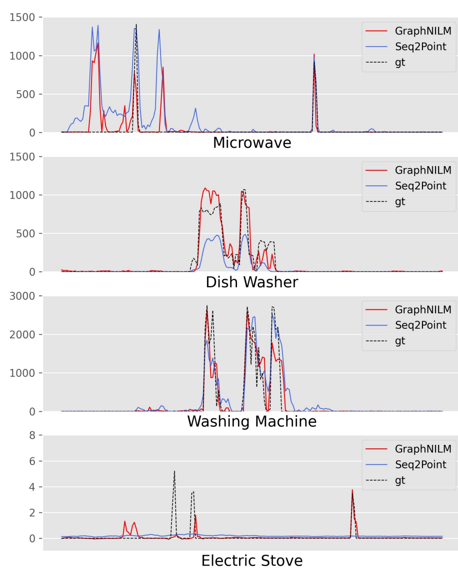

Fig. 3. REDD House 1 case study.

The REDD dataset contains 6 residential houses with 17 uniquely labeled appliances in Boston area from April 2011 to Jun 2011. Though it recorded the power consumption at very low rate, extending up to fifty seconds in some cases, we still choose the training data from 2011-04-20 to 2011-04-30 and test from 2011-05-01 to 2011-05-03 for classic dataset results comparison. **The UK-DALE dataset** contains 5 residential houses with 62 load-level unique labels in Southern England from November 2012 to January 2015. The experiment elected to utilize data from house 1 due to the superior number of appliances and data points collected every six seconds for each appliance therein. The training data is from 2014-02-01 to 2014-02-14 for the completeness of data during this period; the testing data is from 2014-02-15 to 2014-02-28. DTW relations are calculated from 2014-02-01 3 am to 2014-02-07 3am.

4.2 Metrics

For evaluating the performance, MAE and NEP metrics were chosen since they are the most encountered metrics to assess the disaggregated energy [12]. Mean Absolute Error (MAE): MAE measures how accurately the disaggregated energy is compared to the true energy consumption. Normalized Error in Assigned Power (NEP) is an accuracy measures across different appliances.

$$\mathrm{NEP} = \sqrt{\sum_{i=1}^{N}\sum_{t=1}^{T}(\hat{y}_t^i - y_t^i)/\sum_{i=1}^{N}\sum_{t=1}^{T}(y_t^i)}. \tag{5}$$

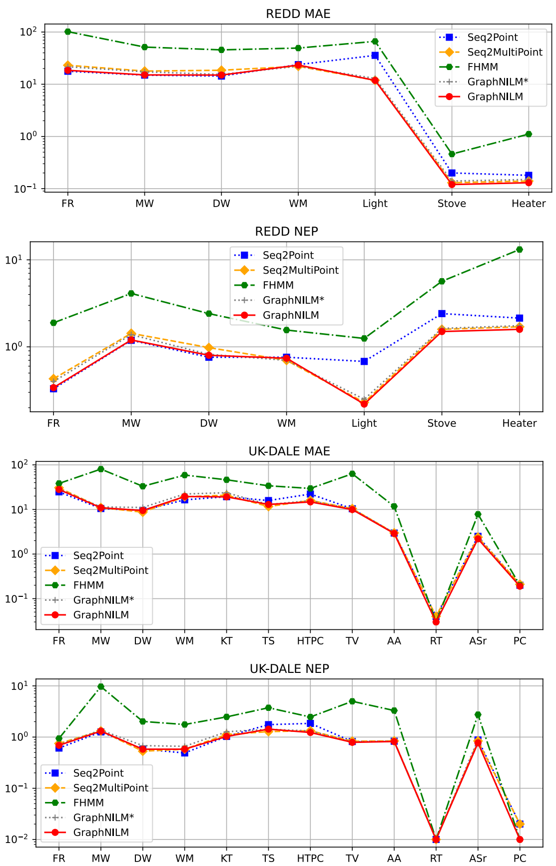

Fig. 4. Comparison w.r.t. MAE and NEP (log y-axis). (left to right) Fridge, Microwave oven, Dish washer, Washing machine, Kettle, Toaster, Audio amplifier, Router, Active subwoofer, Computer.

When the requirement is to compare performance across different appliances, the NEP provides a more effective measurement framework. The larger the MAE and NEP values are, the more error is produced by the model compared to the ground truth.

4.3 Results

Figure 3 shows the disaggregation results for 4 devices in REDD, revealing how GraphNIML captures the patterns for different appliances as Seq2Point model does.

REDD. Follow the methodology discussed earlier, seven appliances in REDD dataset were chosen with the results in Fig 4. Overall the proposed model produces comparable results compared to BM model when we only use 3.9% of memory compared to BM deep learning models. For classic method FHMM which uses least of memory resources, its performance is much worse than all the deep learning models. The proposed model has 4 best performance appliances while BM model has 3 best performance appliances. Not surprisingly, Seq2Point model shows slightly better MAE and NEP on disaggregating fridge and microwave, whose pattern could be more easily to learn from separate single model. GraphNILM performs closely to Seq2Point models on these three loads. With load relationship taken into consideration, washing machine, light, stove and heater results from the proposed model outperform results from the BM models. To understand the impact of the DTW and GCN layers in GraphNILM, it is compared to Seq2MultiPoint: GraphNILM is better in 6 out of 7 appliances in MAE and NEP, which means the GraphNILM solution is constantly providing extra information needed to diaggregate energy consumption. The first order difference introduced in the proposed design also contributes to the overall better performance by comparing GraphNILM and GraphNILM*.

UK-DALE. The same methodology is adopted here and 12 appliances are chosen in UK-DALE dataset. The result is shown in Fig 4. In this experiment, the proposed method achieves the same level of performance compared to BM model while only uses 2.6% of memory. Still, Seq2Point produces the best results for fidge and microwave while GraphNILM better disaggregates all other appliances except the washing machine in this dataset. One reason for the better disaggregation in GraphNILM should be the strong asynchronous relationship among loads observed by DTW: the owner of the house usually uses kettle and toaster in a timely order, and computer, audio amplifier, router, active subwoofer are always functioning in nearby time slot. Therefore, these loads with strong relationships converted to graph edge weights in GraphNILM seem to help it outperform other models. When comparing Seq2MultiPoint to the GraphNILM where the only difference is the DTW and GCN layers, the proposed model performs better on 11 out 12 devices in both MAE and NEP, which stresses the importance of the DTW and GCN in the proposed design. With both datasets' results, the proposed model shows a good and reliable performance in general for solving NILM problems. The popular SOA solution Seq2Point trains a dedicated model for

Load amounts	**1**	**4**	**8**	**12**
Seq2Point	3.6 MB	14.4 MB	28.8 MB	43.2 MB
GraphNILM	832 KB	832 KB	832 KB	832 KB

Table 1. Memory usage of Seq2Point and GraphNILM

each chosen appliance separately hence the total memory size increases linearly along with the number of appliances increases. However in GraphNILM since the number of parameter increase is only nodes and edges in the distilling part, the memory increase is much less significant compared to Seq2Point. Table 1 shows the memory usage for Seq2Point and GraphNILM at window size is 99. The number of parameter in Seq2Point is calculated using NILMTK provided model. To disaggregate one device, Seq2Point requires 3.6 million parameters while the proposed GraphNILM model uses 832 thousands parameters. In a modern home, at least 5 appliances are presented to be disaggregate. In this way more than 80% of the energy consumption could be explained. GraphNILM model only requires 5.2% parameters compared to Seq2Point to disaggregate 5 devices. Besides, in UK-DALE experiment, the runtime for GraphNILM is 98.32s while Seq2Point requires 254.74s for 12 loads training. Therefore, in terms of memory saving, efficiency and transfer implementation, GraphNILM shows competitive advantage.

Transferability This paper also did a quick study on the transferability of our model by using our trained model from UK-DALE House 1 to predict UK-DALE House 5. The chosen houses have similar amounts and categories of loads, which is more like the office building usecase. Table 2 shows the proposed model is at least one sigma better than the Seq2MultiPoint model, stressing the DTW and GCN importance again in the proposed design.

Model	FR	MW	DW	WM	KT	TS	HTPC	TV	AA	RT	ASr	PC	Overall
GraphNILM	**43.15**	30.04	22.73	**57.20**	**16.61**	5.64	**63.17**	24.20	**23.61**	**6.03**	4.03	**13.07**	**309.53±13.99**
Seq2MultiPoint	44.37	**29.83**	**20.87**	66.56	18.68	**5.33**	70.22	**20.84**	25.10	6.03	**2.75**	13.08	323.72±13.99

Table 2. UK-DALE House 1 Model transferred to House 5 under metric 3*MAE

5 Conclusion

GraphNILM outperforms the benchmarks in terms of both the total memory saving, runtime efficiency and the overall MAE performance. Especially for loads with evident relationships, such as the TV, toaster, and kettle groups, the proposed method produces nearly all better results than the current state-of-art method. Even for an independent working device like a fridge or washer, GraphNILM achieves comparable satisfactory results based on MAE and NEP. Given the proposed framework only consumes up to the reciprocal of the total load amount of the memory size in the benchmark, the computational cost of a house with typical loads is drastically reduced. Therefore, extensive experiments conducted on REDD and UK-DALE demonstrate the extraordinary competitiveness of less memory usage and better performance provided by GraphNILM. The deployment of the NILM technique in edge devices for commercial use seems to be around the corner.

References

1. Armel, C., Gupta, A., Shrimali, G., Albert, A.: Is disaggregation the holy grail of energy efficiency? the case of electricity. Energy Policy **52**, 213–234 (01 2013)
2. Batic, D., Tanoni, G., Stankovic, L., Stankovic, V., Principi, E.: Improving knowledge distillation for non-intrusive load monitoring through explainability guided learning. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5 (2023)
3. Batra, N., Kelly, J., Parson, O., Dutta, H., Knottenbelt, W., Rogers, A., Singh, A., Srivastava, M.: Nilmtk: An open source toolkit for non-intrusive load monitoring (07 2014)
4. Çimen, H., Çetinkaya, N., Vasquez, J.C., Guerrero, J.M.: A microgrid energy management system based on non-intrusive load monitoring via multitask learning. IEEE Transactions on Smart Grid **12**(2), 977–987 (2020)
5. De Baets, L., Ruyssinck, J., Develder, C., Dhaene, T., Deschrijver, D.: Appliance classification using vi trajectories and convolutional neural networks. Energy and Buildings **158**, 32–36 (2018)
6. Dong, H., Wang, B., Lu, C.T.: Deep sparse coding based recursive disaggregation model for water conservation. In: Twenty-Third International Joint Conference on Artificial Intelligence (2013)
7. Faustine, A., Mvungi, N.H., Kaijage, S.F., Kisangiri, M.: A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem. CoRR **abs/1703.00785** (2017)
8. Ferraz, F.C., Monteiro, R.V.A., Teixeira, R.F.S., Bretas, A.S.: A siamese cnn + knn-based classification framework for non-intrusive load monitoring. Journal of Control, Automation and Electrical Systems

9. Ghosh, S., Chatterjee, A., Chatterjee, D.: Extraction of statistical features for type-2 fuzzy nilm with iot enabled control in a smart home. Expert Systems with Applications **212**, 118750 (2023)
10. Hart, G.W.: Prototype nonintrusive appliance load monitor. Tech. Rep. Progress Report 2, MIT Energy Laboratory (Sep 1985)
11. He, K., Stankovic, V., Stankovic, L.: Building a graph signal processing model using dynamic time warping for load disaggregation. Sensors **20**, 6628 (11 2020)
12. Huber, P., Calatroni, A., Rumsch, A., Paice, A.: Review on deep neural networks applied to low-frequency nilm (04 2021)
13. Kelly, J., Knottenbelt, W.: 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. pp. 55–64 (2015)
14. Kelly, J., Knottenbelt, W.J.: 'uk-dale': A dataset recording UK domestic appliance-level electricity demand and whole-house demand. CoRR **abs/1404.0284** (2014)
15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2016)
16. Kolter, J., Johnson, M.: Redd: A public data set for energy disaggregation research. Artif. Intell. **25** (01 2011)
17. Kolter, J., Jaakkola, T.: Approximate inference in additive factorial hmms with application to energy disaggregation. International Conference on Artificial Intelligence and Statistics pp. 1472–1482 (04 2012)
18. Liu, B., Zhang, J., Luan, W.: Load oscillation pattern detection for nilm based on scale space decomposition. In: 2023 8th Asia Conference on Power and Electrical Engineering (ACPEE). pp. 1590–1595 (2023)
19. Liu, Y., Xu, Q., Yang, Y., Zhang, W.: Detection of electric bicycle indoor charging for electrical safety: A nilm approach. IEEE Transactions on Smart Grid pp. 1–1 (2023)
20. Mauch, L., Yang, B.: A new approach for supervised power disaggregation by using a deep recurrent lstm network. pp. 63–67 (12 2015)
21. Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio (2016)
22. Parson, O., Ghosh, S., Weal, M., Rogers, A.: Non-intrusive load monitoring using prior models of general appliance types (07 2012)
23. Sykiotis, S., Athanasoulias, S., Kaselimi, M., Doulamis, A., Doulamis, N., Stankovic, L., Stankovic, V.: Performance-aware nilm model optimization for edge deployment. IEEE Transactions on Green Communications and Networking pp. 1–1 (2023)
24. Vine, D., Buys, L., Morris, P.: The effectiveness of energy feedback for conservation and peak demand: A literature review. Open Journal of Energy Efficiency **2**, 2013 (03 2013)
25. Zhang, B., Zhao, S., Shi, Q., Zhang, R.: Low-rate non-intrusive appliance load monitoring based on graph signal processing. In: 2019 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC) (2019)
26. Zhang, C., Zhong, M., Wang, Z., Goddard, N., Sutton, C.: Sequence-to-point learning with neural networks for nonintrusive load monitoring (2016)
27. Zhao, B., Stankovic, L., Stankovic, V.: Blind non-intrusive appliance load monitoring using graph-based signal processing. In: 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP) (2015)