SELF-SUPERVISED LEARNING OF APPLIANCE USAGE

Anonymous authors
Paper under double-blind review

ABSTRACT
Learning home appliance usage is important for understanding people’s activities and optimizing energy consumption. The problem is modeled as an event detection task, where the objective is to learn when a user turns an appliance on, and which appliance it is (microwave, hair dryer, etc.). Ideally, we would like to solve the problem in an unsupervised way so that the method can be applied to new homes and new appliances without any labels. To this end, we introduce a new deep learning model that takes input from two home sensors: 1) a smart electricity meter that outputs the total energy consumed by the home as a function of time, and 2) a motion sensor that outputs the locations of the residents over time. The model learns the distribution of the residents’ locations conditioned on the home energy signal. We show that this cross-modal prediction task allows us to detect when a particular appliance is used, and the location of the appliance in the home, all in a self-supervised manner, without any labeled data.

1 I N T R O D U C T I O N
Learning home appliance usage patterns is useful for understanding user habits and optimizing electricity consumption. For example, knowing when a person uses their microwave, fridge, oven, coffee machine or dish washer provides information about their eating patterns. Similarly, understanding when they use their TV, air-conditioner, or washing machine provides knowledge of their behavior and habits. Such information may be used to encourage energy saving by optimizing appliance usage, to track the wellbeing of elderly living alone (Armel et al., 2013; Donini et al., 2013; Debes et al., 2016), or to provide users with behavioral analytics (Zhou & Yang, 2016; Zipperer et al., 2013).

The problem can be modeled as event detection – i.e., given the total energy consumed by the house as a function of time, we want to detect when various appliances are turned on. Past work has looked at analyzing the energy signal from the home utility meter to detect when certain appliances are on. Most solutions, however, assume that the energy pattern for each appliance is unique and known, and use this knowledge to create labeled data for their supervised models. (Kolter et al., 2010; Zhong et al., 2014, 2015; Kelly & Knottenbelt, 2015; Zhang et al., 2018; Bonfigli et al., 2018). Unfortunately, such solutions do not generalize well because the energy pattern of an appliance depends on its brand and can differ from one home to another (Kelly & Knottenbelt, 2015; Bonfigli et al., 2018). The literature also contains some unsupervised methods, but they typically have limited accuracy (Kim et al., 2011; Kolter & Jaakkola, 2012; Johnson & Willsky, 2013; Parson et al., 2014; Wytock & Kolter, 2014; Zhao et al., 2016; Lange & Berges, 2018).

Unsupervised event detection in a data stream is intrinsically challenging because we do not know what patterns to look for. In our task, not only may appliance energy patterns be unknown, but also the energy signal may include many background events unrelated to appliance activation, such as the fridge or HVAC power cycling events.

One way to address this challenge is to consider the self-supervised paradigm. If a different stream of data also observes the events of interest, we can use this second modality to provide self-supervising signals for event detection. To that end, we leverage the availability of new fine-resolution motion sensors which track the locations of people at home (Adib et al., 2015; Joshi et al., 2015; Li et al., 2016; Ghourchian et al., 2017; Hsu et al., 2017). Such sensors operate as a consumer radar, providing decimeter-level location accuracy. They do not require people to wear sensors on their bodies, can operate through walls, and track people’s locations in different rooms.

These location sensors indirectly observe the events of interest. Specifically, they capture the change in user locations as they reach out to an appliance to set it up or turn it on (put food in a microwave and turn it on). Hence, the output of such sensors can provide a second modality for self-supervision.

1 The utility meter outputs the sum of the energy of all active appliances in a house as a function of time.
2 For example, a Samsung dishwasher may have a different energy pattern from that of a Kenmore dishwasher.
But how should one design the model? We cannot directly use location as a label for appliance activation events. People can be next to an appliance but neither activate it nor interact with it. We also cannot use the two modalities to learn a joint representation of the event in a shared space. This is because location and energy are unrelated most of the time and become related only when the event of interest occurs. Furthermore, there are typically multiple residents in the home, making it hard to tell which of them interacted with the appliance.

Our model is based on cross-modal prediction. We train a neural network that, given the home energy at a particular time, predicts the location of the home residents. Our intuition is that appliance activation events have highly predictable locations, typically the location of the appliance. In contrast, background energy events (power cycling of the fridge) do not lead to predictable locations. Thus, our model uses this learned predictability along with the associated location and energy representation to cluster the events in the energy stream. In addition, we formulate a mixture distribution to disentangle irrelevant location information from other residents in the home. Interestingly, our model not only learns when each appliance is activated but also discovers the location of that appliance in the home, all without any labeled data.

We summarize the contributions of this paper as follows:

- The paper introduces a new method for self-supervised event detection from weakly related data streams. The method combines neural cross-modal prediction with custom clustering based on the learned predictability and representation. We apply it to the task of detecting appliance usage events using unlabeled data from two sensors in the home: the energy meter, and a location sensor.
- To evaluate our design, we have created the first dataset with concurrent streams of home energy and location data, collected from 4 homes over a period of 7 months. For each home, data was collected for 2 to 4 months. Ground truth measurements are provided via smart plugs connected directly to each appliance.
- Compared against past work on unsupervised learning of appliance usage and a new baseline that leverages the two modalities, our method achieves significant improvements of 66.7% and 48.8% respectively for the average detection F1 score.

We will release our code and dataset to encourage future work on multi-modal models for understanding appliance usage patterns and the underlying user behavior.

2 RELATED WORK

Energy disaggregation Our work is related to past work on energy disaggregation, which refers to the problem of separating appliance-level energy from a home’s total (or aggregate) energy signal. Past work in this domain can be broadly classified into two categories: supervised and unsupervised.

Supervised methods assume that the power signatures of individual appliances are available. They use data from individual appliances to obtain models for each appliance power signature, and then use those models to detect appliance events from the aggregate energy signal. Early work learns sparse codes for different appliances (Kolter et al., 2010) or uses a Factorial HMM (FHMM) (Ghahramani & Jordan, 1996) to model each appliance as an HMM (Zhong et al., 2014; 2015). More recently, neural networks have been used to model appliances (Kelly & Knottenbelt, 2015; Zhang et al., 2018; Bonfigli et al., 2018), where extracting appliance-level energy is formulated as a de-noising problem. However, supervised solutions do not generalize well to new homes (Kelly & Knottenbelt, 2015; Bonfigli et al., 2018). This is because two appliances of the same type (e.g. coffee machine) in different homes are often manufactured by different brands, and thus have different power signatures.

Unsupervised methods do not assume prior knowledge of appliance signatures; they attempt to learn those signatures from the aggregate energy signal. Early work uses variants of FHMM, and learn appliance HMMs with Expectation-Maximization (Kim et al., 2011), approximate footprint extraction procedures (Kolter & Jaakkola, 2012), or use expert knowledge to configure prior parameters (Johnson & Willsky, 2013; Parson et al., 2014). Some papers propose using contextual information (such as temperature, hour of the day, and day of the week) (Wytock & Kolter, 2014), or use event-based signal processing methods to cluster appliances (Zhao et al., 2016). More recently, Lange & Berges (2018) proposed using a recurrent neural network as the variational distribution in learning the FHMM. In contrast, our work leverages people’s location data as a self-supervising signal. We cluster appliance events through learning the relation between energy events and people’s locations, and also learn appliance locations as a by-product.
Passive location sensing Motivated by new in-home applications and continuous health monitoring, recent years have witnessed an increasing number of indoor location sensing systems (Adib et al., 2015; Joshi et al., 2015; Li et al., 2016; Ghourieman et al., 2017). They infer people’s locations passively by analyzing how people change the surrounding radio signals (e.g. WiFi) and do not require people to wear any sensors. These sensors have been used for various applications including activity recognition (Wang et al., 2014; 2015), sleep monitoring (Zhao et al., 2017), mobility tracking (Hsu et al., 2017), and health monitoring (Kaltiokallio et al., 2012). In our work, we leverage the availability of such sensors to introduce location data as an additional data modality for learning appliance usage patterns.

Self-supervised multi-modal learning Our work is related to a growing body of work on multi-modal learning. Most approaches learn to encode the multi-modal data into a shared space (Harwath et al., 2018; Owens & Efros, 2018; Zhao et al., 2018). In contrast, since our two modalities are mostly unrelated and become related only when an activation event happens, we learn to predict one modality conditioned on the other. Our work is also related to cross-modal prediction (Krishna et al., 2017; Owens et al., 2016; Zhang et al., 2017) but differs from it in an essential way. Past work on cross-modal prediction typically uses the prediction as the target outcome (e.g. output text for video captioning). In contrast, our objective is to discover the hidden appliance activation events. Thus, we design our method to leverage the learned predictability and cross-modal mapping for clustering activation events. Furthermore, we introduce a mixture prediction design to disentangle unrelated information in our predicted modality (location measurements unrelated to energy events).

3 Problem Formulation

Our goal is to learn appliance activation events in an unsupervised way, using two input streams: home aggregate energy and residents’ location data. Figure 1 shows the two data modalities. We describe each of them formally and define appliance “events” below.

Aggregate energy signal A household’s total energy consumption is measured by a utility meter regularly. This measures the sum of energy consumption from all appliances at each point in time. We denote the aggregate energy signal by \( y = (y_1, y_2, \ldots, y_T) \), where \( y_t \in \mathbb{R}_+ \). Suppose there are a total of \( K \) appliances in a home, and each appliance’s energy signal is denoted by \( x_k = (x_{1,k}, x_{2,k}, \ldots, x_{T,k}) \), where \( x_{t,k} \in \mathbb{R}_+ \). Only the aggregate energy signal is observed, \( y_t = \sum_{k=1}^{K} x_{t,k} + \epsilon_t \), where \( \epsilon_t \sim \mathcal{N}(0, \sigma^2) \) is the background noise.

Figure 1(a) shows one day of an aggregate energy signal. The base power level shifts constantly throughout a day, depending on the background load (e.g. ceiling lights). Added on top of the base level are the various appliance events. Figure 1(b) zooms in around 20:30, and shows examples of those events. The stove was turned on around 20:28, and its power continues to cycle between a few levels. While the stove was on, the microwave was also turned on and ran for a few minutes, and the garbage disposer was turned on shortly.
Indoor location data  We use a single location sensor similar to that in [Hsu et al. 2017] to measure people’s indoor locations passively. The sensor sends out radio signals and analyzes the reflections to localize multiple people. Similarly to a regular WiFi router, the sensor has a limited coverage area of up to 40 feet. Suppose there are \( P \) people in the coverage area at time \( t \). The location data is denoted by \( l_t = (l_{t,1}, l_{t,2}, \ldots, l_{t,P}) \), where \( l_{t,p} \in \mathbb{R}^2 \) is the x-y location at time \( t \) of person \( p \). We can represent the location data over multiple time frames as \( l_{1:T} = (l_1, l_2, \ldots, l_T) \). Figure 1c shows one minute of location data from two people, and Figure 1d shows the data from a top-down view.

Appliance activation events  When an appliance is turned on, it causes a jump in energy consumption, i.e. a leading edge in the energy signal, as shown in Figure 1c. We call such a pattern an appliance activation event. On the other hand, when an appliance changes its internal state, it can also cause a change in the energy signal as shown in the same figure. We call such a pattern a background event. We are interested in discovering activation events to learn appliance usage patterns. Thus, for each jump in the aggregate signal, we take a time window (default is 25 seconds) centered around that jump, and analyze it to detect whether it is an activation event and which appliance it corresponds to.

4 MODEL

Our model operates on time windows (25 seconds) centered around jumps in aggregate energy signal, and the corresponding time windows of location data. The model aims to detect appliance activation events by finding windows with highly predictable user locations conditioned on the energy signal.

Figure 2 shows our model. The idea underlying our model is to first learn a representation of appliance events by finding windows with highly predictable user locations conditioned on the energy signal. This is achieved through the appliance energy encoder \( E \). We can then use the appliance type to predict the location data through the location predictor \( L_e \), which is conditioned on \( z_{t,cat} \). Since people’s locations have information unrelated to appliance events, the total location predictor is a mixture of \( L_e \) and a second module \( L_g \) which captures event-independent location information. Below, we describe these modules in detail.

Appliance Energy Encoder  Given a window of aggregate energy signal \( y_{t:t+w_1} = (y_{t}, y_{t+1}, \ldots, y_{t+w_1}) \) the encoder \( E \) encodes the series into an event vector \( z_t \). We break the event vector into two parts: a categorical vector \( z_{t,cat} \) and a continuous vector \( z_{t,cont} \). We aim to capture the appliance type with \( z_{t,cat} \) (e.g. microwave vs. dishwasher), and use \( z_{t,cont} \) to capture the variability within an appliance. A softmax layer is applied to \( z_{t,cat} \) to ensure that it is a valid distribution over appliance types. \( E \) is parametrized using convolution layers, with one fully connected layer to produce \( z_{t,cont} \) and another for \( z_{t,cat} \). We denote by \( \theta_E \) the parameters of the encoder.

Location predictors  We try to predict the location data conditioned on the appliance event, i.e. we predict a window of locations \( l = l_{t:t+w_2} = (l_t, l_{t+1}, \ldots, l_{t+w_2}) \) centered around the appliance event. We handle multiple people’s locations with a mixture model. Specifically, we use \( L_e \) to predict...
We design a density-based clustering algorithm leveraging the cross-modal relation we learned. Our choice
where \( p \) at the time when a person interacted with the appliance. For events not related to human activities (e.g.,
we blur each location measurement with a Gaussian kernel on the x-y plane to create an image, and process the window of locations \( l_{t,t+w_2} \) frames of images. We reuse the notation \( I \) to represent frames of location images \( \in \mathbb{R}^{[X] \times [Y] \times [T]} \), where \( [X], [Y], [T] \) are the number of discretized points on the x-y and time dimensions. By presenting location data as images, we also remove the variable \( P_t \) while handling a variable number of people in each frame.

We choose \( p_{\theta_{L_e}}(l|z_{t,cat}) \) to be a multivariate Gaussian with a diagonal covariance structure:
\[
p_{\theta_{L_e}}(l|z_{t,cat}) = \mathcal{N}(l; \mu_e, \Sigma) = \prod_{x,y,t} \mathcal{N}(l_{x,y,t}; \mu_{x,y,t}, \sigma_{x,y,t}^2)
\]
where \( \mu_e = L_e(z_{t,cat}; \theta_{L_e}) \) and \( \sigma_{x,y,t} \) to be a constant. We use 3D deconvolution networks to model \( L_e \), which takes \( z_{t,cat} \) as input and outputs the means of the location distributions. We model \( p_{\theta_{L_g}}(\cdot) \) and \( L_g \) in a similar way.

During training, given a window of data \( (l^{(i)}, y_{t:t+w_1}^{(i)}, c^{(i)}) \), we minimize the negative log likelihood of the mixture distribution in predicting the locations:
\[
L_{loc}(\theta_E, \theta_L; l^{(i)}) = -\log p_{\theta_{L_e}}(l^{(i)}|y_{t:t+w_1}^{(i)}, c^{(i)})
\]
Note that as the gradient flows through \( z_{t,cat} \), the likelihood is a function of both \( \theta_E \) and \( \theta_L \). Hence, the encoder \( E \) also learns to encode the energy series based on the concurrent location data.

**Energy Decoder** The decoder \( D \) takes both \( z_{t,cat} \) and \( z_{t,cont} \) and learns to reconstruct the original input energy series by predicting \( y_{t:t+w_1} \). We minimize the reconstruction loss during training:
\[
L_{rec}(\theta_E, \theta_D) = ||y_{t:t+w_1} - y_{t:t+w_1}||_2
\]
The reconstruction loss encourages the encoder \( E \) to produce good initial vectors for \( L_e \) to predict locations. At the same time, it serves as a regularizer to prevent encoder \( E \) from generating meaningless vectors by overfitting location predictions.

**Training** We train all components to jointly optimize the location predictions and energy reconstructions. We minimize the total loss:
\[
L_{total} = \sum L_{loc} + \lambda \cdot L_{rec}
\]
The network implementation and training details are discussed in Appendix 8.4.

### 4.1 Clustering Appliance Events with Cross-Modal Predictions

Once the model is trained, we obtain for each window of energy data its appliance event vector \( z_{t,cat} \) and its cross-modal location prediction \( p_{\theta_{L_e}}(\cdot|z_{t,cat}) \). Next, we use these two vectors for clustering. We design a density-based clustering algorithm leveraging the cross-modal relation we learned. Our intuition is that activation events for the same appliance will cluster together since they have the same appliance type and the same location. We omit the \( cat \) notation below for brevity.

It is typically difficult to cluster in a space learned by a neural encoder because the transformation is highly non-linear and the distance metric is not well-defined. We circumvent this problem by associating the encoded space with a Euclidean space, in which we can easily measure distance. Specifically, for two event vectors \( z_1 \) and \( z_2 \), we can measure their distance in the location space using \( p_{\theta_{L_e}}(\cdot|z_1) \) and \( p_{\theta_{L_e}}(\cdot|z_2) \).

The location prediction \( p_{\theta_{L_e}}(\cdot|z_i) \) represents the likelihood of observing any location \( l_{x,y,t} \) around the time of the appliance event. We found that for events related to human activities (e.g., turning on a kettle or microwave), \( p_{\theta_{L_e}}(\cdot|z_i) \) shows a peak value at the location of the appliance in the x-y space at the time when a person interacted with the appliance. For events not related to human activities (e.g. fridge cycles or random background events), \( p_{\theta_{L_e}}(\cdot|z_i) \) has low values and is diffused.

We define the location predictability score (or the confidence of location prediction) as \( s(z_i) = \max_{x,y,t} p_{\theta_{L_e}}(l_{x,y,t}|z_i) \), and the location distance \( D_{loc} \) between two events as:
\[
D_{loc}(z_1, z_2) = ...
\]

\(^4\)We choose \( \lambda \) to be 0.1 in our experiments to put more emphasis on the location prediction.
\begin{align*}
\|(x_1^* - x_2^*, y_1^* - y_2^*)\|_2, \text{ where } (x_1^*, y_1^*, t_1^*) = \arg\max_{x,y,t} p_{\theta_{\mathcal{E}}} (x, y, t | z_i). \text{ Similarly, the neighborhood distance } D_{\text{eb}} \text{ between two events is defined as } D_{\text{eb}}(z_1, z_2) = \|(z_1 - z_2)\|_2.
\end{align*}

Our clustering algorithm starts with a \(z_i\) with high predictability score \(s(z_i)\). It expands the cluster around \(z_i\)'s local neighborhood in the \(z\) space. It stops expanding if a neighbor’s location distance \(D_{\text{loc}}\) is too far from the cluster center. If all neighbors of the current cluster are visited and none has a small enough \(D_{\text{loc}}\), we start a new cluster from another event with high predictability score. The algorithm is described formally in Algorithm 1.

### Algorithm 1: Clustering energy events with the learned cross-modal relations

**Input:** \(\{z_i\}\) and \(s(\cdot)\): event vectors and their location predictability scores
\(\eta_{\text{eb}}, \eta_{\text{loc}}, \eta_{z}\): thresholds for predictability score, location distance, neighborhood distance
\(N_{\text{min}}\): the minimum number of samples to form a valid cluster

**Output:** Clusters of appliance activation events that are associated with a consistent location

\begin{algorithmic}
\State \(Z \leftarrow \{z_i | s(z_i) > \eta_{z}\}\)
\Function{EL-SCAN}{\(\eta_{\text{eb}}, \eta_{\text{loc}}, \eta_{z}, N_{\text{min}}\)}
\While{\(Z \neq \emptyset\)}
\State \(z_{\text{seed}} \leftarrow \arg\max_{z} s(z_i)\)
\State \(\text{clusters} \leftarrow \{z_{\text{seed}}\}\)
\State ExpandCluster\(k, z_{\text{seed}}, \eta_{\text{eb}}, \eta_{\text{loc}}, \eta_{z}\)
\EndWhile
\State \textbf{return} clusters with at least \(N_{\text{min}}\) examples
\EndFunction
\Function{ExpandCluster}{\(k, z, \eta_{\text{eb}}, \eta_{\text{loc}}, \eta_{z}\)}
\State \(z_{u_k} \leftarrow \text{compute current cluster center}\)
\State \(Z_{\text{nb}} \leftarrow \{z_i \in Z | D_{\text{eb}}(z_i, z) < \eta_{z} \text{ and } D_{\text{loc}}(z_{u_k}, z) < \eta_{\text{loc}}\}\) \Comment{Find valid neighbors}
\State \(Z \leftarrow Z \setminus Z_{\text{nb}}\)
\State \(\text{clusters} \leftarrow \text{clusters} \cup Z_{\text{nb}}\)
\State Repeat ExpandCluster\(\) for all \(z_i\) in \(Z_{\text{nb}}\)
\EndFunction
\end{algorithmic}

### 5 Dataset
We collected concurrent streams of aggregate energy signal and location data from 4 homes over 7 months. We use this dataset for our evaluation. To obtain ground truth labels of appliance events, we deployed programmable smart plugs on the power outlet associated with each appliance. Since not all appliances can be measured by a smart plug (some appliances do not connect to a power outlet), we also developed a labeling tool for manual labeling. The tool allows labelers to label appliance events from the aggregate energy signal, with the help of smart plug data and information collected from the home residents. The choice of sensors and their sampling rates are detailed in Appendix 8.1.

### 6 Results
We evaluate our model and clustering algorithm on unsupervised appliance activation event detection and their learned appliance locations.

#### 6.1 Unsupervised Appliance Event Detection
For appliance event detection, we compare with four baselines. Our method and two baselines have access to location information. EL-Kmeans takes both energy and location data as input and directly clusters them using k-means (Arthur & Vassilvitskii [2007]). E-only-Kmeans clusters only the energy signal with k-means. Methods with location information pre-filter the events and discard events without any location data, as they are unlikely to be activation events. The other two baselines only take the total energy signal as input: AFAMAP (Kolter & Jaakkola [2012]) uses factorial HMM, and VarBOLT (Lange & Berges [2018]) uses a recurrent neural network to model aggregate appliance signals. We use publicly available implementations for these methods (implementation [40]).

We use the same hyper-parameters for the network architecture, training, and clustering algorithm across all homes. As our clustering algorithm is non-parametric, we choose the same number of clusters that it discovers for other methods if possible. For VarBOLT, we report results using 10 clusters, since the training time grows exponentially with the number of clusters and training with more clusters is prohibitively slow. As in past unsupervised work, we report the detection F1 scores based on the best cluster assignments with the ground truth appliances.


[40] We refer to the implementation of VarBOLT for its hyper-parameters.
Table 1 shows that our method has an average detection F1 score of 73.2%, outperforming other baselines ranging from 4.1% to 24.4%. As reported by Bonfigli et al. (2018), AFAMAP performs better when appliance-level data is available for training the HMMs. In the unsupervised setting, however, its footprint extraction procedure does not always produce meaningful HMMs for individual appliances (Bonfigli et al., 2018; Beckel et al., 2014), causing degraded performance. VarBOLT’s training objective focuses on explaining the total amount of energy in a home. Thus, it often uses multiple components (clusters) to model appliances that are on for a long period (e.g. fridge, heater, and dryer/washer). These types of appliances generate many background events, making the algorithm focus less on activation events of other appliances.

Comparing our method with baselines that also have location information (E-only-Kmeans and EL-Kmeans), our approach still outperforms them significantly. E-only-Kmeans performs better than AFAMAP and VarBOLT, showing that the presence of location data is highly related to activation events. However, naively using the location data for clustering does not improve the results, as EL-Kmeans performs similarly to E-only-Kmeans. This is because not all location data is related to appliance events and vice versa. Our approach “cleans up” the data by learning the relation between the two modalities and discovers clusters with strong cross-modal predictability.

Table 2 shows a break down of our results for different appliances.

### Table 1: Unsupervised Appliance Event Detection. Averaged F1 scores (%) of all appliances.

<table>
<thead>
<tr>
<th>N_{appliances}</th>
<th>Ours</th>
<th>EL-Kmeans</th>
<th>E-only-Kmeans</th>
<th>AFAMAP</th>
<th>VarBOLT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home 1</td>
<td>8</td>
<td>82.3</td>
<td>34.8</td>
<td>23.7</td>
<td>6.1</td>
</tr>
<tr>
<td>Home 2</td>
<td>8</td>
<td>69.7</td>
<td>21.6</td>
<td>21.7</td>
<td>5.7</td>
</tr>
<tr>
<td>Home 3</td>
<td>6</td>
<td>76.2</td>
<td>17.3</td>
<td>22.2</td>
<td>5.2</td>
</tr>
<tr>
<td>Home 4</td>
<td>6</td>
<td>64.4</td>
<td>23.9</td>
<td>27.1</td>
<td>9.1</td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>73.2</td>
<td>24.4</td>
<td>23.7</td>
<td>6.5</td>
</tr>
</tbody>
</table>

### Table 2: Unsupervised Appliance Event Detection. Our method’s F1 score (%) for each appliance.

<table>
<thead>
<tr>
<th></th>
<th>Home 1</th>
<th>Home 2</th>
<th>Home 3</th>
<th>Home 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kettle</td>
<td>91.9</td>
<td>-</td>
<td>-</td>
<td>98.6</td>
</tr>
<tr>
<td>Hair dryer</td>
<td>88.0/98.3</td>
<td>-</td>
<td>-</td>
<td>1.1</td>
</tr>
<tr>
<td>Coffee machine</td>
<td>96.1</td>
<td>75.8</td>
<td>90.4</td>
<td>-</td>
</tr>
<tr>
<td>Microwave</td>
<td>81.9</td>
<td>82.1</td>
<td>88.1</td>
<td>96.7</td>
</tr>
<tr>
<td>Stove-activation</td>
<td>90.6</td>
<td>88.1</td>
<td>-</td>
<td>92.7</td>
</tr>
<tr>
<td>Disposer</td>
<td>62.5</td>
<td>78.5</td>
<td>53.1</td>
<td>-</td>
</tr>
<tr>
<td>Toaster</td>
<td>-</td>
<td>48.6</td>
<td>71.1</td>
<td>-</td>
</tr>
<tr>
<td>Blender</td>
<td>-</td>
<td>9.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Door / leaving</td>
<td>-</td>
<td>96.1</td>
<td>83.4</td>
<td>-</td>
</tr>
<tr>
<td>Iron</td>
<td>-</td>
<td>-</td>
<td>71.1</td>
<td>-</td>
</tr>
<tr>
<td>Rice Cooker</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.1</td>
</tr>
<tr>
<td>Others</td>
<td>49.4</td>
<td>76.7</td>
<td>-</td>
<td>96.3</td>
</tr>
<tr>
<td>Average</td>
<td>82.3</td>
<td>69.7</td>
<td>76.2</td>
<td>64.4</td>
</tr>
</tbody>
</table>

### Table 3: Ablation study.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Avg. F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Ours</td>
<td>73.2</td>
</tr>
<tr>
<td>2 Learned embeddings + K-means</td>
<td>60.0</td>
</tr>
<tr>
<td>3 Remove L_q</td>
<td>69.1</td>
</tr>
<tr>
<td>4 Remove L_c &amp; L_q</td>
<td>32.9</td>
</tr>
</tbody>
</table>

6.2 Ablation Study

We perform an ablation study to show our results are contributed by all components in our method. As shown in Table 3, we compare our clustering algorithm (Method 1) with a different algorithm that concatenates the learned multi-modal embeddings (z_{t,cat} and \( p_{\theta}(z_{t,cat}) \)) and directly clusters them with k-means (Method 2). Our clustering algorithm is more effective than directly clustering the multi-modal embeddings, providing an improvement of 13.2% in the average F1 score. This is because our clustering algorithm treats the two modalities differently. For location predictions, we can leverage our understanding of physical distance to set cluster boundaries. For the energy embedding, since it is a non-linear mapping with no clear distance metric, we adopt a density propagation algorithm that only uses local neighborhood distance.

Apart from our clustering algorithm, we evaluate the benefits of our mixture component L_q by experimenting with removing L_q from the model, which reduces the F1 score by 4.1% (Method
1 vs 3). This shows the importance of having $L_q$ extract background motion to allow the location predictor, $L_c$, to focus on modeling the person who interacts with the appliance.

We also consider removing both $L_q$ and $L_c$, and clustering the input based only on the energy embedding $z_{c,cat}$ since there is no learned location predictions. The results shown under Method 4 demonstrate the importance of the location embedding generated by the combination of $L_q$ and $L_c$.

### 6.3 Learned Appliance Locations

Our model also learns the locations where people interacted with appliances, which are typically close to the appliances’ physical locations (we discuss remotely activated appliances in Appendix 8.6). For each appliance event, we take the location predicted by $L_c$ with the highest predictability score, and compare that with the ground truth appliance location measured by a laser meter. The average location prediction error is 0.77 meters with a standard deviation of 0.48 meters across homes. Figure 4a shows the location predictions and their ground truth of several appliance events in Home 1. The corresponding energy signals are shown in Figure 4b–4e.

The location information helps disambiguate appliances with similar energy signals. For example, although the hair dryer and kettle (Figure 4d and Figure 4e) have very similar energy signatures, their different locations (green and orange in Figure 4a) guide the model to encode their events differently.

![Figure 4: Energy signals of discovered activation events and their learned locations from $L_c$.](image)

### 6.4 Location Predictions of $L_c$ vs $L_q$

We visualize the location predictions from the event-related predictor $L_c$ and the event-independent predictor $L_q$ to illustrate how they handle scenarios with multiple people. Figure 5 shows how the mixture design handles the two types of locations. Since $L_c$ is conditioned on energy events, it naturally learns to predict locations related to appliance events. In this case, the location of the hair dryer is predicted by $L_c$ (Figure 5b). On the other hand, $L_q$ predicts the typical locations people tend to stay (e.g., the couch in Figure 5c) based on the context. Having $L_q$ to explain the other locations helps $L_c$ focus on learning the event-related locations.

![Figure 5: Observed locations and predictions of $L_c$ and $L_q$ at a given time for a hair dryer event.](image)

### 6.5 Contextual Location Information and Cluster Visualizations

In Appendix 8.2, we discuss emerging contextual relations between indoor locations through learning cross-modal predictions. We also visualize the learned event vectors to shed light on the design rationales behind our clustering algorithm in Appendix 8.3.

### 7 Conclusion

We introduced a self-supervised solution for learning appliance usage patterns in homes. We infer appliance usage by learning from data streams of two modalities: the total energy consumed by the home and the residents’ location data. Our approach improves on unsupervised appliance event detection significantly, and learns appliance locations and usage patterns without any supervision.
REFERENCES


Sensible Pi. Open energy monitor [https://openenergymonitor.org/](https://openenergymonitor.org/)


Variational BOLT implementation. [https://github.com/INFERLab/varbolt](https://github.com/INFERLab/varbolt)


Hyungsul Kim, Manish Marwah, Martin Arlitt, Geoff Lyon, and Jiawei Han. Unsupervised disaggregation of low frequency power measurements. In *Proceedings of the 2011 SIAM international conference on data mining*, pp. 747–758. SIAM, 2011.


TP-link. Tp-link smart plug hs110 [https://www.tp-link.com/uk/home-networking/smart-plug/hs110/]


8 Appendix

8.1 Sensors Details

In this section, we describe details of the sensors used in our dataset collection.

**Aggregate energy signal** For flexible data collection, we install a sensor (emonPi) at the main circuit breaker in each house as a proxy for the utility meter. We programmed the sensor to collect the raw aggregate energy signal at 1.2 kHz. We down-sampled the data to 10 Hz for our problem to emulate the achievable data rate from a utility meter hardware (Armel et al., 2013).

**Location data** The wireless location sensor is built on a design similar to Hsu et al. (2017). It is a single stand-alone sensor that hangs on the wall, and passively collects multiple people’s locations with decimeter-level accuracy. We down-sampled the location streams to 1 Hz.

**Appliance-level data (for ground truth labeling)** We use TP-Link smart plugs (TP-link) with energy monitoring features for collecting appliance-level data. We wrote custom software using available APIs to collect appliance energy signals at 1 Hz. For appliances that cannot be connected to a smart plug, we asked the residents to write down appliance usage times to help with manual labeling.

8.2 Contextual Location Information via Learned Appliance Events

By analyzing the location predictions of $L_c$ conditioned on different appliance events, we also discover interesting contextual relations between different indoor locations. Figure 6 visualizes the location predictions at different frames around a kettle event. We plot the per-frame location predictability score (or prediction confidence) over time in Figure 6a. The score peaks around $t = 0$s, the time of the event. This is because when people turn on a kettle, they may approach it from different locations, but the location when they push the button is consistent and can be predicted confidently. As a result, the prediction at $t = 0$s correctly shows the kettle’s location (Figure 6d).

Interestingly, a smaller peak of predictability score shows at $t = -10$s in Figure 6a. If we look at the location prediction from $t = -10$s to $t = 0$s (Figure 6b - Figure 6d), we see how the prediction moves from the sink to the kettle. This is because people often fill water at the sink before starting the kettle. Through learning the cross-modal relation, contextual information among locations also emerges as different appliance events are discovered.

![Figure 6: Visualizing location predictions at different times conditioned on a kettle event.](image)

8.3 Visualization of the Learned Event Vectors and Location Predictability

To illustrate what the model learns and the design rationales behind our clustering algorithm, we visualize the space of the learned event vectors $z_{t,cat}$ and their location predictability score $s(z)$. Figure 7 shows the t-SNE (Maaten & Hinton, 2008) visualization of the event vectors on a 2-dimensional space. We color coded the events with three metrics: location predictability scores (Figure 7a), cluster ID discovered by our algorithm (Figure 7b), and ground truth label (Figure 7c). The predictability score depends on how strongly an appliance event co-occurred with a particular
location. As shown in Figure 7a most appliances related to human activities have high predictability
scores (e.g., kettle, hairdryer, microwave, coffee machine, etc). On the other hand, appliances that
cycle in the background (e.g., heater) have very low predictability. The stove has many clusters of
background events. This is because when the stove is on, it cycles between a few power levels, and
the cycle durations depend on the heating levels. Interestingly, we found that stove clusters with
higher power levels (“stove-big-cycle”) also have high predictability scores, while others with cycling
states (“stove-cycle”) show low scores. This is likely because people are next to the stove more often
when the heating level is high.

We can also see that without clustering using both location predictions and event vectors, it is hard to
separate some of the cluster boundaries. Besides, learning to relate energy events to location data
enables us to measure the distances of events in a well-defined physical space.

Figure 7: t-SNE visualization of the learned event vectors colored by (a) location predictability scores
(b) discovered clusters, and (c) ground truth labels.

8.4 Network Implementation and Training Details

In this section, we provide implementation and training details of our neural network model. We
use convolution and deconvolution layers for the energy encoder and decoder. Each module has 8
layers with a kernel size of 3 and a stride of 2. We choose the dimensions of $z_{t,cat}$ and $z_{t,cont}$ to
be 128 and 3. The location predictors have 5 layers of 3D deconvolution with a kernel size of 3
and a stride of 2 in each dimension. The frames of location images for each time window have 32
× 32 × 32 pixels. We discretize the x, y, and time dimensions into 32 points, where the range of
the x-y dimensions are 10 meters and the time dimension has 32 seconds. The neural networks are
implemented in Tensorflow (Abadi et al., 2016). For training, we use the Adam (Kingma & Ba, 2014)
optimizer with a learning rate of 0.001 and a batch size of 64.

8.5 Clustering Parameters

In all experiments, we set $\eta_{D_{loc}} = 0.4$ meters, $\eta_z = 0.03$, $\eta_s = 0.2$, and $N_{min} = 10$. These values are
chosen based on physical and computational constraints. The value of $\eta_{D_{loc}}$ is based on the minimum
physical separation between two appliances. The value of $\eta_z$ only affects the search space in each
iteration, and is chosen to be small for computational efficiency. The minimum predictability score $\eta_s$
is chosen based on a validation set from one of the homes. The value of $N_{min}$ is set to 10 to say that
we need the appliance to appear in the data at least 10 times before we trust that it is a real appliance.

8.6 Remotely Activated Appliances

From our experience collecting the dataset, the vast majority of the appliances used on a daily basis
(Table 2) require human interaction. For example, a person has to put food into a microwave before
turning it on, to hold a hair dryer while drying hair, and to push a button to get a coffee machine
running. Even for an appliance with a remote controller, as long as the person has a regular place
to interact with the appliance from (e.g., always turning the TV on while sitting on the couch), our
model can still learn to predict the location of interaction.