Relation Inference among Sensor Time Series in Smart Buildings with Metric Learning

Shuheng Li,[1] **Dezhi Hong,**[2] **Hongning Wang**[3]
[1]Peking University, Beijing, China
[2]University of California, San Diego, CA USA
[3]University of Virginia, Charlottesville, VA USA
shuhengli@pku.edu.cn, dehong@ucsd.edu, hw5x@virginia.edu

Abstract

Smart Building Technologies hold promise for better livability for residents and lower energy footprints. Yet, the rollout of these technologies, from demand response controls to fault detection and diagnosis, significantly lags behind and is impeded by the current practice of manual identification of sensing point relationships, e.g., how equipment is connected or which sensors are co-located in the same space. This manual process is still error-prone, albeit costly and laborious.

We study relation inference among sensor time series. Our key insight is that, as equipment is connected or sensors co-locate in the same physical environment, they are affected by the same real-world events, e.g., a fan turning on or a person entering the room, thus exhibiting correlated changes in their time series data. To this end, we develop a deep metric learning solution that first converts the primitive sensor time series to the frequency domain, and then optimizes a representation of sensors that encodes their relations. Built upon the learned representation, our solution pinpoints the relationships among sensors via solving a combinatorial optimization problem. Extensive experiments on real-world buildings demonstrate the effectiveness of our solution.

Introduction

Smart Building Technologies hold great potential for improving residents' comfort while reducing energy footprints (Schumann, Ploennigs, and Gorman 2014). These technologies, from demand response controls to fault detection and diagnosis, require the knowledge about the sensing and control points in a building, including what they measure, where they are located, how they are connected, and more. However, this contextual information about each point is represented as *metadata* (as shown in Fig. 1) following vendor-specific naming conventions, and therefore varies significantly in vocabulary and structure from one building to another. Consequently, a necessary first step in deploying any smart building application would be to obtain the contextual information about the points in the building.

However, currently, obtaining this information is a costly, laborious process that often involves domain experts or technicians visiting the site and manually inspecting the points (Dong and Lam 2014). This process can take weeks to complete, and the need for repeating is not necessarily eliminated, as buildings are often retrofitted or renovated. Because new equipment can be installed or new walls can be set up, the context of points will change and thus the metadata requires updates. Simply investing more man-hours is neither scalable nor economical; we need an automated solution to obtain the sensor context.

Figure 1: An example of building metadata and the corresponding encoded contextual information.

In this study, we are particularly interested in two kinds of fundamental relationships as illustrated in Fig. 2: (1) *functional* relationship — which Variable Air Volume (VAV) Box is connected to which Air Handling Unit (AHU), and (2) *spatial* relationship — which sensors are co-located in the same physical space. These relationships provide key context for many analyses. For example, to detect overheated rooms for energy savings, one needs to know the room in which a temperature sensor is located and the corresponding temperature setpoint.

Recent advances have been made in sensor context inference, including the type (Balaji et al. 2015; Gao, Ploennigs, and Berges 2015; Hong et al. 2015), location (Hong et al. 2013; Koc, Akinci, and Bergés 2014; Koh et al. 2016), as well as relationships between sensors (Smith, Sookoor, and Whitehouse 2012; Pritoni et al. 2015). However, these works either build upon problem-specific knowledge that does not generalize (Smith, Sookoor, and Whitehouse 2012; Hong et al. 2013), or still involve a human in the loop, thus error-prone and not scalable (Bhattacharya et al. 2015; Pritoni et al. 2015; Koh et al. 2016). By contrast, the method proposed in this paper will determine the functional and spatial relationships between points with *minimal* manual setup and configuration effort, and thus scale much better.

The key insight behind our solution is that, as two pieces of equipment are connected or a group of sensors is co-

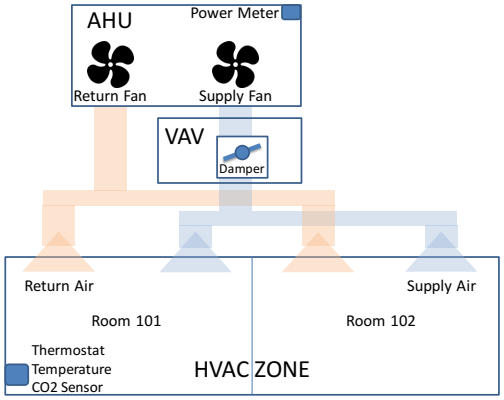

Figure 2: A typical building contains multiple air handling units (AHUs), each heating/cooling the air and circulating it to dozens of variable air volume (VAV) boxes. Each VAV fine tunes the airflow for a single room, where multiple sensors are instrumented to monitor its temperature, humidity, CO2 level, etc. Building analytics require the relations among the vast amounts of data streams (e.g., which VAV connects to which AHU, and which sensors are in the same space), which are currently acquired via a costly, laborious, yet still error-prone manual process.

located, they are exposed to the same real-world events, e.g., a fan turning on or a person entering the room, thus exhibiting correlated changes in their sensor reading time series. These correlated changes in turn reflect the relation between the sensor time series. However, it is highly non-trivial to realize a solution following this intuition, due to two major technical challenges. First, the event-triggered patterns in sensor data are not necessarily synchronous, e.g., a change in an AHU would take a longer time to affect VAVs afar than a nearby one, and the resulting changes are distinct, e.g., room temperature changes much faster than CO_2 concentration. Consequently, to explicitly correlate events from these sensor readings involves solving a complex segmentation and matching problem — an event can span over an indeterminate number of readings starting at an arbitrary timestamp in different sensor streams. To circumvent this complexity of explicitly finding and correlating events, we convert the raw time series readings to the frequency domain, where event-triggered changes are characterized by different frequency bands, mitigating the effect of misalignment or shifts in time series. With these transformed signals, we propose a deep metric learning method that directly learns a non-linear feature representation of sensor streams, which implicitly encodes the correlated events for relation inference.

Once we have the representation, relation inference in a (possibly large) set of time series is yet another challenging combinatorial optimization problem to solve. The fundamental principle guiding our solution is that a sensor should best correlate with the others in relation due to functional connection or physical co-location. Therefore, to identify the relationships we search for a grouping among sensors such that it maximizes the total sum of the intra-group pairwise similarity between sensors for all the groups produced. Since exhaustive search for all the possible groupings is intractable, we appeal to different approximate optimization algorithms for efficiency and accuracy trade-off.

We perform an extensive evaluation using data from seven office buildings consisting of tens of thousands of sensing points with millions of readings in total, and demonstrate the effectiveness of our solution in identifying the two relationships of interest. To the best of our knowledge, this is the first attempt to facilitate the relation inference process for sensors at such a scale. We believe the proposed method is promising and could potentially apply to a broader set of relation inference for sensors.

Related Work

Our work is related to two bodies of research — sensor relational inference and metric learning for time series.

Sensor Relation Inference. Efforts on standardizing resource organization and management (e.g., sensors, devices, equipment, etc) in smart buildings are emerging, including a uniform schema (Balaji et al. 2016) and methods and tooling (Koh et al. 2018) on inferring the type (Balaji et al. 2015; Gao, Ploennigs, and Berges 2015; Hong et al. 2015) and location (Hong et al. 2013) of sensors, as well as relations among them (Koc, Akinci, and Bergés 2014; Pritoni et al. 2015). Our work is particularly concerned with relation inference.

There are two different approaches to acquiring relationships between sensors: parsing the sensor metadata (i.e., point names as shown in Fig. 1) and inferring from the sensor time series readings. Bhattacharya et al. (Bhattacharya et al. 2015) developed a programming language-based approach to automatically parse the sensor names and obtain the relationships between sensors. Schumann et al. (Schumann, Ploennigs, and Gorman 2014) used string matching with a manually created dictionary to find the meaning of sensor labels and derive the relationships. While effective, however, these approaches can work well **only if** the point names are available and the relations of interest are encoded in the names, which is not always the case in practice.

There have also been recent efforts in the latter category, i.e., identifying relationships from the sensor readings time series. Hong et al. (Hong et al. 2013) showed that, by removing dominant diurnal patterns from the raw sensor readings, they can identify co-located sensors with decent accuracy. Koc et al. (Koc, Akinci, and Bergés 2014) measured linear correlation to infer the spatial relationships between discharge air sensor and zone temperature sensor in a room. However, while promising, the results in these studies are obtained using only a dozen sensors from a handful of rooms. Pritoni et al. (Pritoni et al. 2015) discovered the functional relationships between AHUs and VAV boxes by perturbing the operation of AHUs and leveraging the responses in VAVs. Similarly, Koh et al. (Koh et al. 2016) perturbed the VAV control parameter and can identify the sensors installed in the same VAV. However, these approaches take weeks to execute and require domain-specific knowledge about when and how to perturb operations in a way that does not interfere with building needs, which does not generalize to other kinds of relationships. There are also approaches that do not

rely on perturbations. Park et al. (Park, Lasternas, and Aziz 2018) developed a data-driven solution that cross-correlates the raw measurements from a particular pair of sensors in the equipment and takes the majority match over a period of time. Hong et al. (Hong et al. 2019) proposed to directly infer the "events" in equipment using their sensory time series data and correlate based on these events to find AHU-VAV functional connections. Our solution also does not require perturbations to system operation and further exploits the phenomena already encoded in the time series data to find *multiple* kinds of relationships between various types of equipment and sensors, requiring minimal manual input.

Similarity Metric Learning for Time Series. One of the most popular similarity measures for time series is Dynamic Time Warping (DTW) (Berndt and Clifford 1994), which computes the best alignment of two time series' indices to minimize the overall aligned distance. Apart from the optimal alignment strategy in DTW, studies have proposed using all-pairs alignment distance to measure the similarity between time series (Yeh et al. 2016; Gharghabi et al. 2018). Based on the concept of alignment distance, deep neural networks are further introduced to obtain a representation of time series data. A recent work builds a Siamese network to extract features and uses an expected all-pairs alignment distance as the similarity metric (Che 2017). To improve the pre-defined global alignment distance, (Grabocka and Schmidt-Thieme 2018) uses deep networks to learn an indicator function for all-pairs alignment. In addition to alignment methods, Recurrent Neural Networks (RNN) are also used to model the similarity between time series (Mueller and Thyagarajan 2016; Pei, Tax, and van der Maaten 2016), where the state of the last hidden layer is directly used as the representation of time series and the similarity is computed using negative $L1$ or $L2$ distance function. However, time series from building sensors is often noisy; small time-shifts, outliers, and highly varied event patterns can result in poor performance for these time-domain similarity metrics (Chan, Fu, and Yu 2003). We instead appeal to a solution that explores the characteristics of sensor data in the frequency domain.

Methodology

Relation inference among sensor time series is non-trivial, as the event-triggered patterns in sensor readings highly depend on the properties of the sensors or equipment, e.g., location and measurement type. Most of the existing approaches that find the asynchronous correlation between time series are composed of two components: a method to align time series that warp non-linearly on the time dimension, e.g., DTW, and a predefined distance function (Che 2017). Despite abundant improvements, this branch of approaches still bears $O(T^2)$ computational complexity, where T is the length of time series, and suffers from noise as well as outliers in sequences.

To handle the complex correlation problem of time series while circumventing computationally costly time-domain based algorithms, we propose to identify and encode the similarity of time series based on the event-triggered patterns in the frequency domain. As illustrated in Fig. 3, our solution first converts sensor time series into the frequency domain using the Short-Time Fourier Transformation (Daubechies 1990) and then extracts features of Discrete Fourier transformed data via a convolutional neural network. The network is trained to embed relatedness between time series, i.e., deep metric learning. In the rest of this section, we first elaborate on the technical details of the aforementioned two key procedures in our algorithm, and then explain how to infer the relations among sensors based on the learned embedding vectors of sensor time series.

Short-Time Fourier Transformation Operator

As we are dealing with time series from sensors of various types and locations, though they are physically connected or co-located, the event-triggered changes in their time series are not necessarily synchronous, and the resultant patterns often vary in "shapes". Instead, characteristics of these events in the frequency domain provide new perspectives for distinguishing the changes in time series. We resort to the Short-Time Fourier Transformation (STFT), which converts a time-domain sequence with varying frequency into a set of frequency-domain components. Specifically, given a time series sequence $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$ as input, STFT breaks it into chunks using a fix-sized sliding window, and the n^{th} element in the m^{th} chunk is derived as

$$STFT^{(\tau,s)}\{\mathbf{x}\}(m,n) = \sum_{t=1}^{T} \mathbf{x}(t) \cdot \mathbf{w}(t - s \cdot m) \cdot e^{-j\frac{2\pi n}{\tau}(t - s \cdot m)}, \tag{1}$$

where $STFT^{(\tau,s)}\{\cdot\}$ denotes the short-time Fourier transform operator with window size τ and sliding stride length s, and $\mathbf{w}(t)$ is a sliding window function with width τ that only has non-zero values for $1 \leq t \leq \tau$. Since the input $\mathbf{x}$ is a real-valued sensor reading time series, each Discrete-Fourier transformed chunk is conjugate symmetric, i.e., only the first $\lfloor \tau/2 + 1 \rfloor$ coefficients of the frequency components are non-redundant.

Intuitively, given a real-world event, the response time of a sensor can be captured by the phase of a sinusoidal wave, and the response duration can be described by the amplitudes of frequency components. Since the events could happen at an unpredictable pace, it is thus challenging to reveal the complex non-linear dependencies between the Fourier coefficients and different types of sensor relations. We appeal to a deep metric learning technique, which we will explain shortly, to recognize the relation dependent feature vectors that represent the relatedness among sensors.

We further process the Fourier coefficients to facilitate the representation learning for neural networks. Specifically, we only preserve the $k \leq \lfloor \tau/2 + 1 \rfloor$ coefficients from the 2^{nd} to the $k+1^{th}$ and discard the rest. The reasons are two-fold: First, the last coefficients correspond to the relatively high-frequency components of the signal, which usually are the noise in the time series data. Extracting the first coefficients corresponds to deriving a sketch of the signal so that the real-world events are preserved while high-frequency noise is removed. Second, the direct current (DC) component, which is the amplitude of 0 Hz term, should also be removed:

$$STFT^{(\tau,s)}\{\mathbf{x}\}(m,0) = \sum_{t=1+s\cdot m}^{\tau+s\cdot m} \mathbf{x}(t). \tag{2}$$

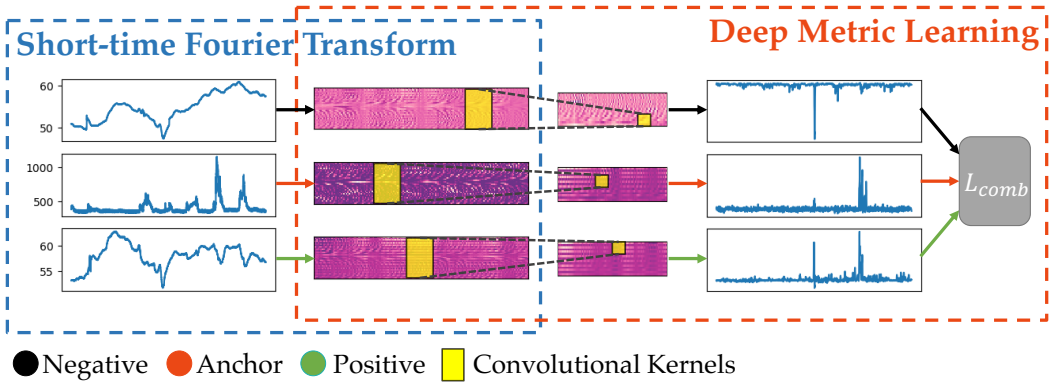

Figure 3: The proposed STFT Triplet Network: A triplet composed of primitive sensing signals — from top to bottom: a negative point, an anchor point, and a positive point — is sampled. The primitive signals are first converted to frequency-domain multi-channel tensors by STFT and then fed to a deep metric learning network. Through multiple convolutional layers, the neural network embeds the input tensors into 1-D feature vectors, which encode the relatedness among the sensing time series.

The DC component (Eq. (2)) can be easily derived from Eq. (1) by setting $n = 0$, and it is equivalent to summing up all the samples in the window. Since the signals are collected from sensors with different types and locations, the DC component only reflects the characteristics of the sensors rather than the events. Therefore, we choose to eliminate it.

Note that the Discrete-Fourier transformed frequency coefficients are complex-valued, while the input to the neural networks in the next step needs to be real-valued. We thus transform the complex-valued coefficients into a real-valued tensor in advance by re-arranging the k coefficients within a chunk into $2k$ frequency channels:

$$a_1, b_1, a_2, b_2 ..., a_k, b_k,$$

where a_n and b_n constitute the n^{th} complex coefficients c_n, as $c_n = a_n + jb_n$ for each $1 \leq n \leq k$. Finally, we derive a 2-D tensor $\mathbf{X} \in \mathbb{R}^{F \times N}$ with $F = 2k$ frequency channels, whose length N is equal to the number of chunks.

Deep Metric Learning Triplet Network

With the primitive sensor reading time series converted to the frequency domain, we seek a means to learn a mapping from the Fourier coefficients to an effective representation of underlying events, through which functionally connected or physically co-located sensors could best correlate with each other. To this end, we design a deep metric learning network that can represent sensors in relation using embedding vectors of a closer distance than those not in relations. It is noteworthy that when looking at each group of sensors in relation, the number of positive samples is much smaller than the number of negative samples (e.g., for co-location inference, the number of sensors in the same room is much smaller than that in different rooms), and thus directly learning the absolute pairwise distance between positive and negative pairs is heavily affected by the unbalanced training data. As a result, we perform metric learning via a triplet network to capture the relative relatedness among sensors instead.

The triplet network is comprised of three identical feed-forward networks with shared parameters. In each iteration, a mini-batch of training triplets consisting of an anchor sensor $\mathbf{X}_a$, accompanied with a pair of positive sensor $\mathbf{X}_p$ (i.e., a sensor in functional/spatial relation) and negative sensor $\mathbf{X}_n$ (i.e., a sensor not in functional/spatial relation) are fed into the triplet network. In one triplet $\mathcal{T}$, $\mathbf{X}_a$ and $\mathbf{X}_p$ are sampled from the group of related sensors (e.g., those in the same room), while $\mathbf{X}_n$ is sampled from the non-related groups. When fed with a triplet, the network outputs the corresponding embedding vectors $\mathbf{y}_a$, $\mathbf{y}_p$, and $\mathbf{y}_n$. The objective of the network is to learn an embedding space such that the anchor sensor is closer to the positive sensor than to the negative sensor, i.e., $d_p = ||\mathbf{y}_a - \mathbf{y}_p||^2 < d_n = ||\mathbf{y}_a - \mathbf{y}_n||^2$.

We achieve this objective by combining the triplet loss (Weinberger and Saul 2009) and the angular loss (Wang et al. 2017). Specifically, the **triplet loss** is defined as,

$$L_{tri}(\mathcal{T}) = \left[||\mathbf{y}_a - \mathbf{y}_p||^2 - ||\mathbf{y}_a - \mathbf{y}_n||^2 + \gamma\right]_+, \quad (3)$$

where $\mathcal{T}$ denotes the input triplet and $\left[\cdot\right]_+$ denotes the hinge loss function. The goal of the triplet loss is to push the negative sensor point away from the anchor by a global distance margin $\gamma > 0$ compared to the positive sensor point.

Accordingly, the **angular loss** is defined as,

$$L_{ang}(\mathcal{T}) = \left[||\mathbf{y}_a - \mathbf{y}_p||^2 - \mu||\mathbf{y}_c - \mathbf{y}_n||^2\right]_+, \quad (4)$$

where $\mathbf{y}_c$ is the mean vector for the anchor and positive sensor points, and μ is a weight parameter. Note that μ was originally described as an angular upper bound $4\tan^2\alpha$ in (Wang et al. 2017), we predigest the format by simply viewing it as a hyper-parameter for simplicity. The main insight of angular loss is to push the negative point away from the local cluster centroid defined by $\mathbf{y}_a$ and $\mathbf{y}_p$.

We shall note that, generally, the triplet loss directly optimizes the relative distance between positive pairs and negative pairs, while the angular loss imposes an additional angular constraint on the triplet. To take advantages of both loss functions, we design a new loss function by introducing a trade-off weight λ between Eq. (3) and Eq. (4) to boost the overall performance:

$$L_{comb}(\mathcal{T}) = L_{tri}(\mathcal{T}) + \lambda L_{ang}(\mathcal{T}). \quad (5)$$

Considering that there is no locality among nearby frequency bands in the input 2-D tensor $\mathbf{X} \in \mathbb{R}^{F \times N}$, as they are supposed to be orthogonal to each other, we only apply convolution along the time dimension created by the moving windows. Therefore, as shown in Fig. 3, the convolutional kernels cover the entire frequency spectrum and move along the time dimension to extract the complex dependency patterns in the input channels. Our convolution networks consist of 4 convolutional layers with ReLU activation and max-pooling layer applied between 2 consecutive layers. It is noteworthy that we use one 1×1 convolutional filter that covers the entire input channels for the last block to generate a 1-D feature vector as the final output of the whole network.

Relation Inference

After obtaining the embedding vectors of the sensor time series data, the final step of our solution is to uncover the relationships between sensors. We consider the task of relation inference as a graph cut problem, where each vertex represents a sensor or equipment and only the sensing points in a given relation should be connected after the cut. We set the weight of each edge to be the similarity between the embedding vectors of two vertices, and relation is obtained via min-cut on this graph. Additional constraints can be added: for example, in spatial inference, the number of connected components (i.e., number of rooms) and number of sensors in each room could be provided ahead of time, i.e., a minimum k-cut problem. This however significantly increases the complexity, as minimum k-cut is NP-complete (Garey and Johnson 2002). Given the difficulty of this combinatorial optimization problem, we appeal to approximated algorithms, e.g., genetic algorithm (Deb et al. 2002) and greedy algorithm, for solutions.

For functional relationship inference, given the precondition that each VAV is connected to only one AHU and that there is no connection within VAVs or AHUs, the problem can be simplified to graph-cut in a bipartite graph setting. Therefore, we use a greedy algorithm to assign each VAV to the AHU with which it has the highest similarity. For spatial relationship inference, we have the constraint that l rooms are given, where each is instrumented with the same t types of sensors, and one sensor for each type. This is a minimum l-cut problem, and each connected component has exactly t vertices. To solve this constrained combinatorial optimization problem, we employ a genetic algorithm to approximate the optimal solution.

Our employed approximation algorithms are task-specific and not dependent on the deep metric learning model. Developing an end-to-end solution that can directly optimize the neural network with respect to relation inference quality would be widely favored; we leave it as our future work.

Implementation Details

Before training, all the sensor reading time series are converted into the frequency domain using the STFT operator. In particular, we use overlapped sliding windows to mitigate the resolution loss at the edges of each window, and use rectangular windows to weigh each time point equally. To make the neural networks robust to the varying event patterns due to different sensor types or locations, we normalize the learnt embedding vectors to a unit length, i.e., $||\mathbf{y}||^2 = 1$. We update the weight of each convolutional kernel by back-propagating the loss defined in Eq. (5) using mini-batch stochastic gradient descent. For the final step, the similarity between two embedding vectors is calculated as $\rho(\mathbf{y}_1, \mathbf{y}_2) = 1 - ||\mathbf{y}_1 - \mathbf{y}_2||^2$, which is equivalent to Pearson Correlation Coefficient when the vectors are normalized.

Empirical Evaluations

Experiment Setup

Datasets. To evaluate our solution for inferring functional and spatial relationships among sensing time series, we use two different real-world datasets. In particular, for functional relation inference, we use the data from 6 large commercial buildings located across the U.S.: the number of AHUs in each building varies from 5 to 13, and the number of VAVs ranges from 100 to 300, where the smallest building has over 1,300 sensing and control points installed. Sensor readings are reported every 15 minutes. For each AHU, the number of connected VAVs ranges from a handful to more than 50. The ground-truth of the VAV and AHU connection is obtained from the vendor of these buildings. For spatial relation inference, eight-day worth of data is collected from the sensors in one office building, consisting of 50 office rooms across 4 floors. Particularly, each room is instrumented with four types of sensors, and one for each type — a CO2 sensor, a humidity sensor, a light sensor, and a temperature sensor. The data from these sensors is recorded every 5 seconds.

Baselines. We compare with two categories of solutions for relation inference. The first three baselines are alternative supervised deep learning methods that create feature vectors to represent the raw sensor streams, with which we derive pairwise similarity between sensors using their corresponding predefined similarity measures.

• **Dynamic Time Warping (DTW):** As a clear competitor against Euclidean distance, DTW (Berndt and Clifford 1994) is used as the similarity measure for embedded feature vectors generated from our deep learning model.

• **Deep Expected Alignment Distance (DECADE):** Instead of computing one single best warping path in DTW, this deep network-based model takes all the possible warping paths and computes the expected alignment distance to make training more efficient (Che 2017).

• **Warping Networks (WN):** As a state-of-the-art solution of deep metric learning for time series, this model uses two connected deep neural networks to encode the raw time series and the optimal warping distance respectively (Grabocka and Schmidt-Thieme 2018).

In addition, we also compare with two *unsupervised* baselines for the relation inference that aim at explicitly extracting events from time series.

• **Hidden Markov Model (HMM):** As a straightforward solution, we apply a $K(= 2)$-state discrete HMM to infer the binary event state in the sensor time series data, and use the event sequences for relation inference. • **Markovian Event Model (MEMO):** It uses a 3-layered switching Markovian model for inferring the event state (Hong et al. 2019), and the

Table 1: VAV assignment accuracy (%) for functional relation inference.

Building ID	10312	10320	10381	10596	10606	10642
Unsupervised						
HMM	18.01	11.50	25.64	21.02	31.59	34.75
MEMO	90.42	88.50	90.43	91.28	92.16	92.28
Supervised						
DTW	88.46	25.46	83.81	91.67	55.17	80.78
DECADE	96.54	77.27	93.33	**99.44**	85.98	96.08
WN	97.31	60.91	95.23	**99.44**	77.93	96.86
TN	97.30	42.73	96.19	93.33	52.18	96.07
SSN	97.69	42.73	95.24	97.78	63.67	93.72
STN	**98.07**	**90.00**	**96.23**	**99.44**	**93.10**	**98.03**

Table 2: Accuracy (%) for spatial relation inference.

	Edge Accuracy	Room Accuracy
Unsupervised		
HMM	12.67	4.00
MEMO	19.00	10.00
Supervised		
DTW	37.27 ± 2.40	14.40 ± 2.94
DECADE	12.47 ± 1.61	2.00 ± 6.00
WN	17.47 ± 2.17	8.00 ± 9.80
TN	25.73 ± 1.94	6.00 ± 9.20
SSN	67.93 ± 8.66	50.20 ± 14.32
STN	$\mathbf{88.61 \pm 2.08}$	$\mathbf{80.00 \pm 3.79}$

inferred event sequence via maximum a posterior inference for each sensor is used for relation inference.

For our proposed **STFT Triplet Network (STN)** model, we also inspect a few variants for comparisons. Specifically, we employ the **Triplet Network (TN)** to study the effect of STFT and **STFT Siamese Network (SSN)** to study the effect of the triplet loss architecture.

Evaluation Metrics. For functional relationship, we evaluate the performance of different methods with regard to the *VAV assignment accuracy*, i.e., how many VAVs are correctly assigned to their connected AHUs. For spatial relationship, we measure 1) *edge accuracy*, i.e., the percentage of the *sensor pairs* that an algorithm predicts to be in the same room is actually in the same room; and 2) *room accuracy*, i.e., the percentage of rooms that are correctly recovered, where a room is considered as correctly recovered if and only if all the four sensors that an algorithm assigns to it are actually from the same room.

Model Setups. Because the sampling frequency and length of sensor time series in the two datasets are different, we use two different sets of parameters for the two inference tasks. For *functional inference*, the duration of entire time series is 10-month ($T = 28,800$), after removing invalid values (e.g., erroneous sensor readings). The window size of STFT is set to 30, the stride of each window is 2, and we select the first $k = 14$ complex Fourier coefficients. For *spatial inference*, each eight-day long sensor time series has $130,000$ readings after periods with missing values trimmed. The window size is accordingly set to 200, the stride is 10, and $k = 32$. The learning rate is fixed to 0.0001 for both datasets, and SGD optimizer is used to update the model. We test all the supervised learning methods with 5-fold cross validation. In particular, for functional inference, there are multiple sensors attached to each equipment, and we need to select a particular pair of sensors, one from AHU and one from VAV, for training. By default, we use *Air Flow Volume* in VAVs and *Supply Fan Speed* in AHUs for evaluation, as the two are known to be physically correlated. To further relieve the domain knowledge required in this task, we also test our model using all possible pairs of sensors and will report the results later. And for spatial inference, we observe that the performance of deep learning models varies moderately for each run. To mitigate the effects of randomness and quantify the range of variation, we repeat all the deep learning models 10 times and report their mean accuracy with stand deviation.

Experiment Results

Relation Inference Quality. The experiment results for functional relation inference and spatial relation inference are reported in Table 1 and Table 2, respectively. For the two unsupervised baselines, HMM-based solution performs poorly as expected, since it simply models the event states based on the raw sensor readings. MEMO is designed for functional inference and it on average performs well across all the buildings. However, this event-based algorithm cannot extract efficacious events for spatial inference, which indicates that its event model is over-specialized, thus unable to recognize events in a different context.

For all the supervised learning baselines, we can see that, apart from one building 10596 where DECADE, WN as well as our model achieve the same result, our proposed STN model outperforms all the other baselines in both tasks. Although most of these supervised solutions are competitive on the majority of the buildings, they do not perform well on building 10320 and 10606, which are the noisiest and difficult ones. Most of the baselines also perform poorly in spatial inference, due to varied reasons. DTW is known to be vulnerable to noise and outliers in data. Specifically, DTW tends to mistakenly align two sequences that are not correlated when too much noise exists. This greatly limits the performance of DTW on building 10320 and 10606, and the performance is only slightly better than random assignment. For the two models based on all-pairs' alignment distance (i.e., DECADE and WN), the neural networks contain a tremendous amount of parameters, thus requiring a large set of data for parameter estimation. However, given the relatively small size of the training set for spatial relations, the performance of these neural networks suffers. Furthermore, their degradation on building 10320 and 10606 implies that they also fail to handle real-world sensor streams with low signal-to-noise ratio.

The comparison between different variants of our STN model explains its improved performance. TN works in the time domain by directly taking the raw time series as input to the triplet network, and its performance on the two difficult buildings drops significantly comparing to STN. In addition, its low accuracy for spatial inference demonstrates

Table 3: VAV assignment accuracy using different sensor pairs. The *left* is the result of our model (STN) and the *right* is the result of MEMO.

VAV \ AHU	Supply AirPress	Supply AirTemp	Supply FanSpeed
AirFlowVolume	97.69/75.22	97.69/33.63	98.07/91.15
DischargeAirTemp	98.21/57.52	99.11/42.48	97.82/36.28
SpaceTemp	93.07/13.27	91.51/15.04	96.05/31.86

Table 4: Functional inference accuracy under one-month v.s. ten-month training data.

	STN	WN	DECADE
One-month	47.27	26.36	30.90
Ten-month	90.00	60.91	77.27
Relative Drop (%)	47.48	56.72	60.01

that frequency-domain features are more informative than raw time series in the time domain. Compared to STN, the performance of SSN indicates that the Siamese architecture is fairly unstable with a large variance. It demonstrates that triplet architecture is more effective, especially for small datasets where we do not have a large amount of training data and for unbalanced datasets where the number of negative pairs is far more than the number of positive pairs. The triplet structure can quadratically increase the amount of training data, thus preventing the model from being overwhelmed by large numbers of negative pairs.

Effect of Sensor Pair Selection. Previous experiments for functional inference are based on the domain knowledge about which two sensors are best correlated, i.e., we know that when the *Supply Fan Speed* changes in an AHU, the connected downstream VAVs will exhibit changes in their *Air Flow Volume*. However, in practice, diverse air conditioning systems could be deployed in different buildings with different sensors available, and thus the aforementioned pair of sensors might not always be available for use. We thus inspect how sensitive our model is to the selection of different sensor pairs, and we compare with the current best unsupervised solution, i.e., MEMO.

As shown in Table 3, for the sensor pairs that are not directly correlated, e.g., room temperature (SpaceTemp) and pressure of supplying airflow (SupplyAirPress), the accuracy of MEMO drops drastically to less than 40% while our model can always maintain a high accuracy of over 90% regardless of the pair of sensors chosen. The result indicates that our model can learn an effective representation of the hidden correlations between time series, even when some of them are not directly dependent.

Cross-Building Learnability. As our algorithm still requires labeled training data to obtain an effective embedding of related sensors, reducing the burden of labeling data is valuable. Furthermore, there will be scenarios where the ground-truth annotation for relations is lacking in a target building. A natural solution is to leverage the data of sensors in relation from other buildings to reduce labeling effort in new buildings. To this end, we conduct experiments to examine if our model is able to learn the characteristics of sensors in relation in a *cross-building* setting: we train our relation inference model using the data from building Y and test it on a different building X. It is noteworthy that this is a challenging task, given the heterogeneity of equipment deployed in different buildings. Yet, we shall demonstrate the advantageous transfer capability of our method, in the face of such heterogeneity.

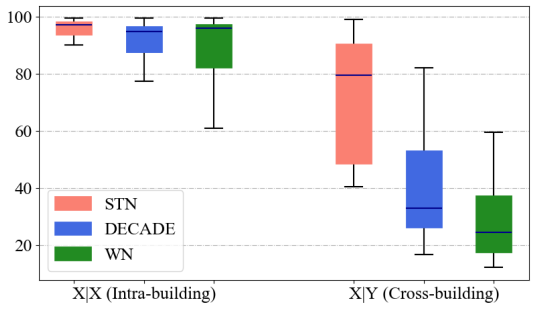

Figure 4: Cross-building inference accuracy for functional relations across all six buildings: '$X|Y$' denotes training on Y and testing on X, and STN is our proposed method.

Fig. 4 illustrates the comparison of our model and two other deep learning baselines – DECADE and WN. We observe only gentle performance degradation for our model with regard to the median accuracy, while the median accuracy of the two baselines models drops significantly. Our model is effective in most scenarios — the first quartile accuracy of our model is significantly higher and our model remains negatively skewed. This is due to the fact that the baseline models merely learn the data-driven representations while our model learns higher-level representations in a task-driven manner (i.e., identifying the underlying event patterns). This distinctive feature allows our model to adapt to a variety of real-world scenarios, e.g., transferring relation inference across buildings.

Effect of Amount of Data. We also investigate how sensitive the models are to the amount of training data. As all baselines generally do not work for spatial relation inference, we compare them in functional relation inference in this experiment. Particularly, we train a model using first-month data and test on ten-month data, and compare the result to the setting when ten-month training data is used. We test on the most difficult building 10320 and compare our model STN with two supervised baselines, DECADE and WN. From Table 4, we see a relatively small performance drop by our model. Yet, admittedly, a loss of nearly 50% in accuracy indicates that further development of our model is required, in order to overcome possible training data scarcity.

Conclusions

In this paper, we develop a deep metric learning solution, combined with approximate search algorithms, to perform relation inference among sensor time series in smart buildings, which currently requires repeating laborious manual

effort. To handle the varying event-triggered patterns across sensor streams, the solution starts from transforming the time-domain readings to the frequency domain, and then appeals to a deep metric learning network to derive an optimized representation of the sensors in relation. Extensive experiment results on several large real-world building datasets demonstrate the effectiveness as well as transfer learning capability of the solution, i.e., it can apply knowledge about related sensors in one building to another.

As future work, we plan to extend our solution to an end-to-end one, where the network would optimize and yield relations directly as the output. It is also valuable to extend the scope beyond smart buildings to more general relation inference problems in sensor networks. We believe this is a promising and important direction, as inferring relations among sensing streams in general could be a critical task in the grand future of Internet-of-Things.

Acknowledgments

This work was supported by National Science Foundation IIS-1718216 and Department of Energy DE-EE0008227.

References

Balaji, B.; Verma, C.; Narayanaswamy, B.; and Agarwal, Y. 2015. Zodiac: Organizing large deployment of sensors to create reusable applications for buildings. In *BuildSys*.

Balaji, B.; Bhattacharya, A.; Fierro, G.; Gao, J.; Gluck, J.; Hong, D.; Johansen, A.; Koh, J.; Ploennigs, J.; Agarwal, Y.; et al. 2016. Brick: Towards a unified metadata schema for buildings. In *BuildSys*. ACM.

Berndt, D. J., and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In *AAAIWS'94*.

Bhattacharya, A. A.; Hong, D.; Culler, D.; Ortiz, J.; Whitehouse, K.; and Wu, E. 2015. Automated metadata construction to support portable building applications. In *BuildSys*.

Chan, F.-P.; Fu, A.-C.; and Yu, C. 2003. Haar wavelets for efficient similarity search of time-series: with and without time warping. *IEEE Transactions on knowledge and data engineering*.

Che, Z. 2017. Decade : A deep metric learning model for multivariate time series. In *KDD Workshop MiLeTS*.

Daubechies, I. 1990. The wavelet transform, time-frequency localization and signal analysis. *IEEE*.

Deb, K.; Pratap, A.; Agarwal, S.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*.

Dong, B., and Lam, K. P. 2014. A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. In *Building Simulation*. Springer.

Gao, J.; Ploennigs, J.; and Berges, M. 2015. A data-driven meta-data inference framework for building automation systems. In *BuildSys*. ACM.

Garey, M. R., and Johnson, D. S. 2002. *Computers and intractability*, volume 29. wh freeman New York.

Gharghabi, S.; Imani, S.; Bagnall, A.; Darvishzadeh, A.; and Keogh, E. 2018. Matrix profile xii: Mpdist: A novel time series distance measure to allow data mining in more challenging scenarios. In *ICDM*.

Grabocka, J., and Schmidt-Thieme, L. 2018. Neuralwarp: Time-series similarity with warping networks. *ArXiv* abs/1812.08306.

Hong, D.; Ortiz, J.; Whitehouse, K.; and Culler, D. 2013. Towards automatic spatial verification of sensor placement in buildings. In *BuildSys*.

Hong, D.; Wang, H.; Ortiz, J.; and Whitehouse, K. 2015. The building adapter: Towards quickly applying building analytics at scale. In *BuildSys*.

Hong, D.; Cai, R.; Wang, H.; and Whitehouse, K. 2019. Learning from correlated events for equipment relation inference in buildings. In *BuildSys*, BuildSys '19. ACM.

Koc, M.; Akinci, B.; and Bergés, M. 2014. Comparison of linear correlation and a statistical dependency measure for inferring spatial relation of temperature sensors in buildings. In *BuildSys*. ACM.

Koh, J.; Balaji, B.; Akhlaghi, V.; Agarwal, Y.; and Gupta, R. 2016. Quiver: Using control perturbations to increase the observability of sensor data in smart buildings. *arXiv preprint arXiv:1601.07260*.

Koh, J.; Hong, D.; Gupta, R.; Whitehouse, K.; Wang, H.; and Agarwal, Y. 2018. Plaster: An integration, benchmark, and development framework for metadata normalization methods. In *BuildSys*. ACM.

Mueller, J., and Thyagarajan, A. 2016. Siamese recurrent architectures for learning sentence similarity. In *AAAI'16*.

Park, J. Y.; Lasternas, B.; and Aziz, A. 2018. Data-driven framework to find the physical association between ahu and vav terminal unit–pilot study.

Pei, W.; Tax, D. M. J.; and van der Maaten, L. 2016. Modeling time series similarity with siamese recurrent networks. *CoRR* abs/1603.04713.

Pritoni, M.; Bhattacharya, A. A.; Culler, D.; and Modera, M. 2015. Short paper: A method for discovering functional relationships between air handling units and variable-air-volume boxes from sensor data. In *BuildSys*. ACM.

Schumann, A.; Ploennigs, J.; and Gorman, B. 2014. Towards automating the deployment of energy saving approaches in buildings. In *BuildSys*.

Smith, V.; Sookoor, T.; and Whitehouse, K. 2012. Modeling building thermal response to hvac zoning. *ACM SIGBED Review*.

Wang, J.; Zhou, F.; Wen, S.; Liu, X.; and Lin, Y. 2017. Deep metric learning with angular loss. *CoRR* abs/1708.01682.

Weinberger, K. Q., and Saul, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* 10.

Yeh, C. M.; Zhu, Y.; Ulanova, L.; Begum, N.; Ding, Y.; Dau, H. A.; Silva, D. F.; Mueen, A.; and Keogh, E. 2016. Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *ICDM*.