Pacific Northwest
NATIONAL LABORATORY

PNNL-30655

# Exploration of Domain Aware Machine Learning for Grid Analytics

## Transfer-Learnt Energy Models to Assist Buildings Control with Sparse Field Data

September 2020

Milan Jain
Khushboo Gupta
Arun Sathanur
Vikas Chandan
Mahantesh Halappanavar (PI)

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

## DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, **makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
*operated by*
BATTELLE
*for the*
UNITED STATES DEPARTMENT OF ENERGY
*under Contract DE-AC05-76RL01830*

**Printed in the United States of America**

**Available to DOE and DOE contractors from the Office of Scientific and Technical Information,**
**P.O. Box 62, Oak Ridge, TN 37831-0062**
www.osti.gov
**ph: (865) 576-8401**
**fax: (865) 576-5728**
**email: reports@osti.gov**

**Available to the public from the National Technical Information Service**
**5301 Shawnee Rd., Alexandria, VA 22312**
**ph: (800) 553-NTIS (6847)**
**or (703) 605-6000**
**email: info@ntis.gov**
**Online ordering: http://www.ntis.gov**

PNNL-30655

# Exploration of Domain Aware Machine Learning for Grid Analytics

## Transfer-Learnt Energy Models to Assist Buildings Control with Sparse Field Data

September 2020

Milan Jain
Khushboo Gupta
Arun Sathanur
Vikas Chandan
Mahantesh Halappanavar (PI)

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

# Contents

# 1.0 Introduction

Building energy modeling is key to the smart grid infrastructure. With rise of innovation in smart grid technologies, quick and accurate prediction of building energy consumption has become even more critical today. Several smart grid programs, such as demand response and curtailment rely on the smooth integration of buildings into the grid infrastructure. However, to ensure unification, these grids often rely on the accurate predictions of buildings' energy consumption in short-term, medium-term, and long-term horizons. The accurate estimation of building energy consumption is even more critical for the building owners, who can use these estimates to plan and utilize their electric appliances, efficiently.

The accurate modeling of building energy consumption relies on various static and dynamic parameters, which includes outside weather conditions, stochastic schedule of the occupants, and appliance usage patterns. While one can easily monitor the building level energy consumption and the weather conditions surrounding the building, it is often hard to sense granular-level building-specific information such as, occupants' schedule and appliance usage patterns. Limited access to such critical information often restricts the deployment of white-box or grey-box models for the building energy modeling. However, prior work has shown promising results with various black-box models (such as Random Forest, Variational Autoencoders, among many others) applied on easy-to-collect data. Though these models can estimate energy consumption with good accuracy, they typically require data for one complete year for training.

In real-world, one cannot wait for one whole year to get the data and then train the model to estimate the building energy consumption. For practical applications, the user would often prefer a plug-n-play solution that can provide accurate predictions starting from day one. Moreover, accuracy is subjective and depends on the requirements of an application. Therefore, if a model trained over data corresponding to a few days, can achieve accuracy numbers comparable to a model trained on a year-long data set, users would typically prefer a quicker solution than the delayed one. In this project, we propose one such plug-n-play building energy modeling framework, which is powered by the concepts of transfer learning and it can estimate the building energy consumption accurately even with the “sparse” field data. Our results indicate that the model can estimate building energy consumption with an accuracy of 68\% with just four days of field data (where the baseline accuracy is 57%) and with 83% with three weeks of data (the baseline accuracy is 71%).

In our present framework, knowledge transfers from the simulation data to the field data, in the form of instance-transfer and model-transfer. While simulation data provides a sense of “generic” behavior of the building, the field data captures the stochastic dynamics of the building. We evaluated the efficacy of our approach on the field data collected from six commercial buildings for a year and our results indicate that transfer-learning based models are much more effective than the baseline models (random forest and feed-forward network), especially when the data is sparse. Our major contributions are:

1. Exploring various transfer learning-based strategies for building energy modeling.
2. Detailed comparison of proposed transfer-learnt models with the state-of-the-art machine learning techniques based on data from six commercial buildings.
3. Plug-n-play transfer-learning based framework for building energy modeling.

# 2.0 Data Exploration

In this section we will discuss about the data and some initial exploration.

## 2.1 Data Collection

### 2.1.1 Simulation Data

The FEDS building energy simulation software tool[1] was chosen based on its ability to generate hourly thermal, electricity, and electric demand profiles of buildings. FEDS uses the input building data and TMY3 weather data to simulate an 8,760-hour model of building system energy use. Specifically, FEDS can take in detailed user input data on building parameters or estimate unspecified or unknown building details from a minimum set of user inputs for analysis that extends from a single building to a large campus. These characteristics include building function, size, age, location, occupancy, geometry, envelope, lighting, HVAC systems and distribution, plug load, and process loads, and marginal utility rates for the energy resources consumed.

FEDS models energy and cost performance of heating, cooling, ventilation, lighting, motors, plug loads, building shell, and service hot water systems from the building parameters and calculates the 8,760 hour energy consumption and electrical demand for each technology, end use, building, and entire campus or installation over a year of location specific TMY3 weather data. The Richland, WA campus of the Pacific Northwest National Laboratory consists of 43 buildings, 10 of which were chosen for the analysis. Hourly load profiles were generated from FEDS models of the selected buildings.

### 2.1.2 Field Data

We collected energy consumption and outside weather conditions data from six commercial buildings for one year. All the six buildings are low-rise buildings with an average occupancy around 200-300 people during the work hours. The work hours in these buildings typically starts at 9 AM and ends at 6 PM. The carpet area of the buildings varies from 2000 sq. ft to 4000 sq. ft. We collected data at 15 minutes interval.

## 2.2 Data Preprocessing

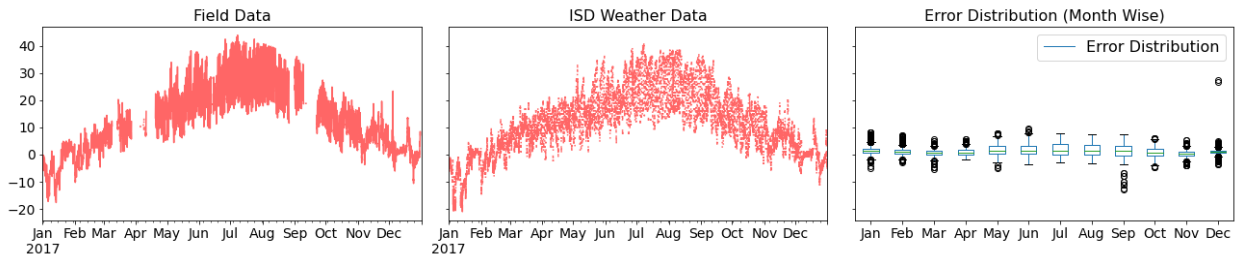


Figure 1 Data Preprocessing: [Left] Temperature Data (Field). [Middle] ISD Temperature Data. [Right] Distribution of error - a mean absolute error of 2°C, across all the buildings.

For a cross-building comparison, we normalized the energy consumption of the building by its total area, also known as Energy Use Intensity (EUI) and expressed as energy per square foot.

[1] https://feds.pnnl.gov/

Besides, we did have some parts of the data set where the outside temperature data was missing. To fill-in those parts, we used weather data from Integrated Surface Database (ISD) from National Centers for Environmental Information (NCEI) of National Oceanic and Atmosphere Administration (NOAA)[1]. As shown in Figure 1, we noticed a mean absolute error of 2°C in the field data and the ISD data, across all the buildings.

## 2.3 Feature Engineering

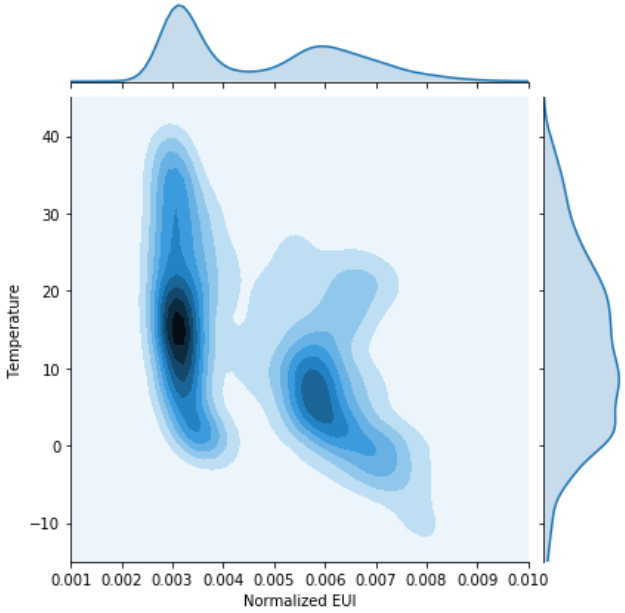


Figure 2 Joint Distribution of Outside Temperature and Energy Consumption (Field Data): The two modalities differentiate non-working hours (left contour) from the working hours (right contour).

Building energy consumption at any time mainly relies on occupants' schedule and the outside weather conditions. Figure 2 shows the correlation between the EUI (which is on the x-axis) and the outside temperature (y-axis) for the field data. The two modalities in the EUI distribution (top x-axis) differentiates the non-working hours (left contour) from the working hours (right contour). One interesting point to note here is the correlation between the energy consumption and the outside temperature in the working hour modality (right contour). The upper half of that contour represents the hot time period of the year and the bottom half of the contour depicts the cold season. The energy consumption is maximum during the peak for both the seasons.

Based on this data exploration, we derived 28 features from the raw data, which are current and previous outside temperature, hour of the day, weekday/weekend, and working/non-working hour. We used one-hot encoding to represent hour of the day - one binary feature vector for each hour. We used a binary vector to denote if it is a weekday or a weekend. We consider Monday-Friday as the weekdays and Saturday-Sunday as the weekend. Likewise, we used a binary vector to identify the working hour and the non-working hour. On any working day, 9 AM-6 PM are the working hours and the others are the non-working hours.

[1] https://www.ncdc.noaa.gov/isd

# 3.0 Transfer Learning

Typically, any machine learning problem statement consists of a domain $D$ and a task $T$. The domain $D$ is the set of feature space $\chi$ and the joint probability distribution $P(X)$, where $X = \{x_1, x_2, \dots, x_n\} \in \chi$. In the case of building energy modeling, at any timestamp t, $x_i$ would depict the feature vector comprising of features like outdoor temperature, week of the day, among many others. For a given domain $D = \{\chi, P(X)\}$, the task $T$ consists of a label space $\gamma$ and an objective predictive function $f_\theta$, where, $\theta$ denotes the trained parameters of the model. In this report, the label space is the energy predictions normalized by the area of the building. To predict the corresponding label, we learn the predictive function on the training examples, that comprises of pairs $\{x_i, y_i\}$, where $x_i \in \chi$ and $y_i \in \gamma$.

As we discussed earlier, the problem with training a predictive function $f_\theta(.)$ on the field data is that we need a "good" amount of data to train a model that can predict the output labels with an "admissible" accuracy. However, one might have to wait for months, if not years, to gather a "good" number of training samples $\{x_i, y_i\}$. If we train a complex model on the sparse field data (cold start), we might end up with the problem of data overfitting - good accuracy on the training set but poor accuracy on the test set, which means that the model cannot generalize well. The long wait for the data can limit the real-world implementation of the data-driven models, and training on sparse data will make them inaccurate. Therefore, instead of cold start, we need a way to train the model and predict energy consumption when we don't have enough data to start. One way to tackle this problem of the *cold start* is to train the model on the simulation data.

In transfer learning, we typically have a source and a target. We define source as the problem statement from which we transfer the knowledge, and target as the problem statement to which we transfer the knowledge. In our case, we define the problem of training a machine learning model on the simulation data as the source problem and training the model on the field data as the target problem. In the rest of this report, we will denote source domain and task as $D_s = \{\chi_s, P(X_s)\}$ and $T_s = \{\gamma_s, f_{\theta_s}(.)\}$, respectively, and target domain and task as $D_t = \{\chi_t, P(X_t)\}$ and $T_t = \{\gamma_t, f_{\theta_t}(.)\}$, respectively. As discussed earlier, the problem with the cold start is that $\gamma_s \neq \gamma_t$, and thus $T_s \neq T_t$. In other words, the distribution of output labels $y$ is different for the simulation data and the field data. When no field data is involved in the training stage, the predictive function $f_{\theta_s}(.)$, trained only on the simulation data, predicts inaccurately because it is unaware of the finer variations in energy consumption in the field data. To resolve this issue, we incorporated the field data in the training stage through instance-based and model-based transfer learning, which we will discuss in detail next.

## 3.1 Instance-Transfer

In instance-based transfer, we assume that the source-domain and the target-domain data use same set of features and labels, but the data distribution in two domains are different. Since both the distributions are different and not all the samples from the source domain are useful, we first optimally sub-sample the training set of the source domain ($X_s$) and then append it to the training instances of the target domains ($X_t$) to reconstruct the training set.

$$opt(X_s) \rightarrow X_s', f_{\theta_t^\circ} \xrightarrow{\{X_s', X_t\}} f_{\theta_t}$$

Equation above depicts the formal representation of the same, where $X_s'$ is the optimally sub-sampled training set from the source domain, $\theta_t^\circ$ is the set of untrained parameters, and $\theta_t$ is the set of trained parameters for the target problem statement.

We used TrAdaBoost to implement the instance-based transfer. TrAdaBoost, automatically and iteratively re-weights the source domain data to reduce the impact of the “bad” source samples and rather focus on the “good” samples.

## 3.2 Model Transfer

Earlier, in the cold start, we directly used the model trained on the simulation data to predict energy consumption for the field data. In model-based transfer, we now retrain the last two layers of the pretrained model using the field data. Equation below presents a formal representation of the model-based transfer.

$$f_{\theta_s} \overset{X_t}{\rightarrow} f_{\theta_t}$$

Here, $f_{\theta_s}$ is the model trained on the simulation data and $f_{\theta_t}$ is the model with last two layers retrained and initial layers fine-tuned with the field data.

# 4.0 Evaluation

We carried out a detailed evaluation to compare transfer-learning methods with the baseline techniques by collecting a year-long data from six commercial buildings.

## 4.1 Baseline Models

For the baseline comparison, we implemented two of the most common used machine learning and deep learning models for building energy modeling - Random Forest and Feed-Forward Network.

### 4.1.1 Random Forest

Random forest in an ensemble learning method that fits several decision trees on various sub-samples of the dataset and averages over them to avoid overfitting and improve the prediction accuracy. We implemented Random Forest in scikit-learn in Python and used Randomized Search for the hyperparameter tuning.

### 4.1.2 Feed Forward Network

Feed Forward Network, also known as multi-layer perceptron, is typically used for supervised machine learning tasks where the target labels are usually known. Formally, the model architecture can be defined as –

$$u^i = relu(W^{n*h} x^i)$$

$$f_1^i = relu(W^{h*h} u^i)$$

$$d^i = dropout(f_1^i)$$

$$f_2^i = relu(W^{h*h} d^i)$$

$$v^i = W^{h*1} f_2^i$$

where for the $i_{th}$ input $x^i$, *relu* is the non-linearity function, $f_1^i$ and $f_2^i$ are the two hidden layers, $W^{n*h}$ is the weight matrix respective to input layer, $W^{h*h}$ is the weight matrix respective to two hidden layers, $W^{h*1}$ is the weight matrix respective to output layer, $n$ and $h$ depict the total number of input variables and hidden nodes, respectively. We have implemented the feed forward network using pytorch and trained it using adam optimizer with varying learning rates. The total number of nodes ($h$) in both the hidden layers was 256 and we mean squared error (MSE) as the loss function.

## 4.2 Evaluation

Given the experiment setup discussed above, we will next evaluate the performance of transfer learning strategies, along with the baseline strategies.

### 4.2.1 Training on Whole Year

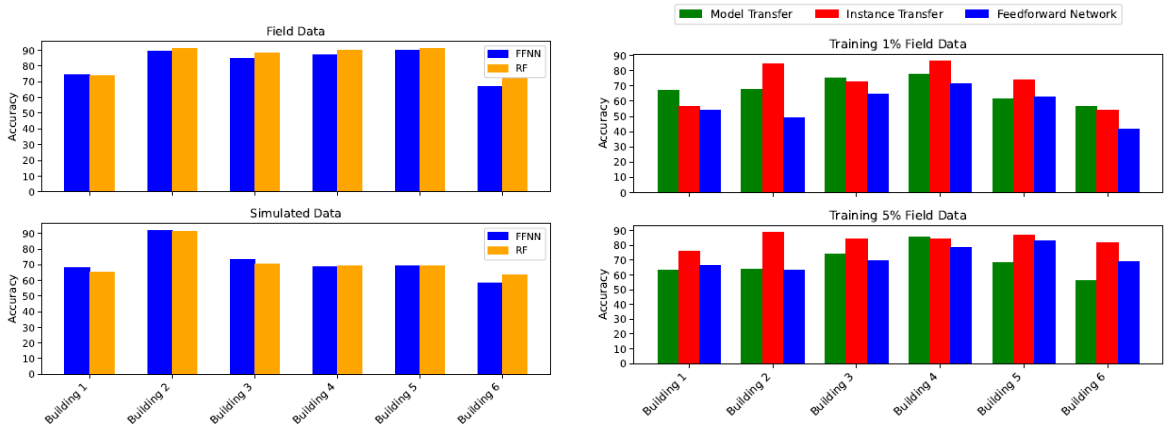


Figure 3 [Left] When trained on one year of data, both the random forest and the feed-forward network had a comparable accuracy in estimating the building energy consumption. [Right] We implemented both transfer learning strategies for the feed-forward network. Our analysis indicates that transfer learning is much more effective than the baseline strategy for the smaller datasets.

We did an 80:20 split for this analysis to generate a baseline accuracy across all the six buildings. We trained two models - Random Forest (RF), Feed Forward Network (FF) - on the 80% random samples of data (approx. 10 months) and evaluated the performance on remaining 20% data (approx. 2 months) for both simulated and field dataset for the six buildings. Our results (as shown in Figure 3) indicate that the prediction accuracy of both feed-forward neural networks and random forest is comparable across all the buildings.

### 4.2.2 Training on Sparse Field Data

The value of transfer learning lies in getting good accuracy with the sparse data. We noticed earlier that the validation accuracy for random forest and feed-forward network drops as we reduce the size of the training data. The drop-in accuracy was the indication of data overfitting. However, we noticed a significant improvement in the prediction accuracy with the transfer learning. As shown in Figure 3, both, instance-based and model-based transfer learning models are much more accurate and consistent, when compared to the baseline strategies. Our analysis, across all the buildings, indicates that transfer learning-based models trained on approx. one week and three weeks of data can predict with better or comparable accuracy than the baseline models.

### 4.2.3 Training of Seasonal Data

In the real-world, the data typically comes in the sequential order. We might not have a few samples of the field data from one whole year, but rather some data from a particular month or the season. As one might notice in Figure 4, the accuracy of the baseline models drop even further when the models are trained on the data samples from a particular season and tested on the remaining seasons. This happens because the model hasn't seen much seasonal variation during the training. However, lack of seasonal variation in the field data makes an insignificant impact on the transfer learning-based methods and the reason being the knowledge gained from the simulation data.

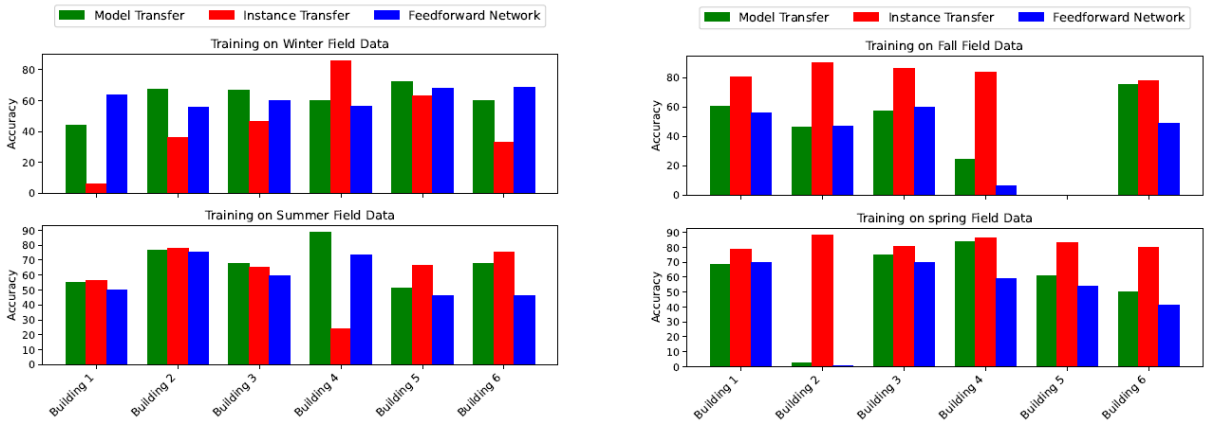


Figure 4 Knowledge transfer is much more effective when the model is trained on the seasonal data. The transfer learning-based methods outperformed the baseline methods with big margins here.

In instance-based transfer, the model is aware of the seasonal variations through random samples of simulated data. Likewise, in the model-based transfer, the knowledge about seasonal variation is embedded in the weight vectors of the pretrained model. This way, even with the lack of field data for other seasons, the transfer learning-based models are doing much better than the baseline models. We did notice some negative transfer in a few cases, especially for the winter season. The impact wasn't evident on the estimation of energy consumption for the fall and the spring, because these seasons are not extreme and there exist some hot and cold days variation. We plan to address these concerns in the future by minimizing the impact of negative transfer.

# 5.0 Discussion

In our analysis, we noticed that transfer-learning based models can perform better than the traditional machine learning and deep learning models, especially when the field data is sparse. This is useful to deal with the problem of cold start, initially, when a building has limited data to start with. With transfer learning, models don't wait for a long time to collect data for the training and they can rather make use of the simulation data. Initially, when a data-driven model doesn't have much data to infer about the seasonal variations and its impact on the energy consumption, the baseline models performed badly. In those cases, our study indicates that knowledge transfer from a model trained from the simulation data to the model that we want to train using the field data can be effective. Since we can generate any number of variations with the help of a simulation model, transfer learning seems to be one of the most promising ways of building energy modeling in the future.

Having said that, we believe this work is only the beginning and needs further exploration. Transfer learning is an active area of research in both machine learning and deep learning. In this project, we implemented and analyzed two most used approaches of transfer learning to solve the problem of cold start. However, this work can easily be extended to solve other data-related concerns for building energy modeling. For instance, we can use transfer learning to tackle the limitation of missing measurements. We can transfer knowledge from the simulation framework to the field data about certain sensors which are not installed in the field, but corresponding measurement is available in the simulation.

An alternate path to build on this work is to explore other variations of transfer learning to boost the performance of the proposed techniques. We did notice a few cases of negative transfer learning (especially when trained on the winter data), that can be avoided by further strengthening the models. In addition to this, one can also study other types of information transfer - such as transfer of feature representation and relational knowledge. In the former one, we aim at finding the “good” feature representation to minimize domain divergence. Likewise, in transfer of relational knowledge, we do not assume that the data drawn from each domain is independent and therefore try to transfer the relationship among data from a source domain to a target domain. A detailed evaluation of the possible variants and methods of transfer learning-based modeling can further help us in understanding, what is the most effective way of knowledge transfer for building energy modeling.

# 6.0 Conclusion

In this project, we implemented and analyzed two transfer-learning based strategies for building energy modeling to tackle the problem of cold-start. For evaluation, we compared the performance of proposed approaches with two state-of-the-art baseline approaches - Random Forest and Feed-Forward Network. Our analysis on the field data collected from six different building for one year indicates that transfer learning models trained on the sparse data can estimate building energy consumption with a comparable or better accuracy than the baseline models. Furthermore, when the data doesn't contain seasonal variations, the transfer learning models are much more effective than the baseline strategies. In the future, we plan to extend this work by further improving the transfer-learning based methods and applying building-to-building transfers.

# Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354
1-888-375-PNNL (7665)

***www.pnnl.gov***