# Conditional Latent Modeling of Electric Vehicle Load Curves

**Oskar Triebe**
Infrastructure Engineering
Stanford University
triebe@stanford.edu

**Will Lauer**
Computer Science
Stanford University
wlauer@stanford.edu

**Tymor Hamamsy**
BMI
Stanford University
tymor@stanford.edu

## Abstract

We explore latent distributions over residential electric load curves and electric vehicle (EV) charging curves and their conditional connection. First we train a Conditional Variational Autoencoder (CVAE) that jointly models the household load and EV curves and add a supervised objective to add or subtract EV charging curves from the household load. Second, we train a dual CVAE, where we model the EV charging curve conditioned on the load curve's latent representation. The learned latent distributions capture meaningful household characteristics. The dual CVAE approach leads to more realistic samples as when sharing model parameters. However, the information captured by a household's general load does not suffice to conditionally generate EV charging curves.

The code associated with the publication has been made public on *github*.

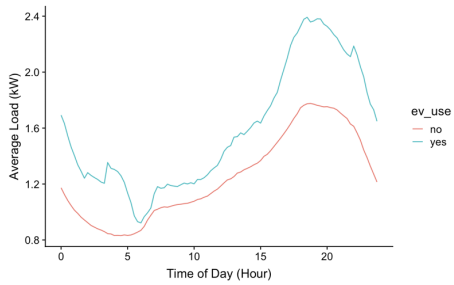
## 1 Introduction

A major problem for grid operators is estimating electrical load curves, which measure the variation in electric load over time. Load profiles vary dramatically between individuals with different electricity demand/usage in different regions. Utility companies, electricity market participants, and governments track how the cumulative development of these profiles impacts the net load on the electric grid. The recent growth of distributed solar energy and electric vehicles (EVs) in residential areas has put significant stress on the electric grid. In order to optimally allocate grid resources it essential for grid operators to estimate future electric load profiles. Figure 1b shows the dramatic ramp up on the grid that occurs at nightfall due to the low net load during daytime due to solar energy and higher evening load due to EVs charging at home.

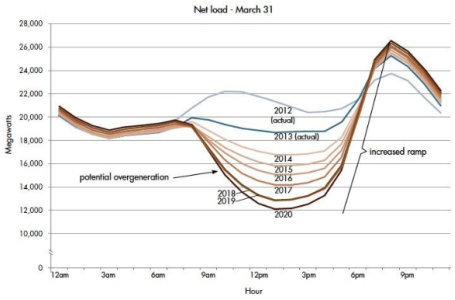
The International Energy Agency projects that the number of EVs on the road will grow from 3 million today to 125 million EVs by 2030 [1]. California proactively supports this and has set a goal to have 5 million low-emission vehicles on the roads by 2030, which is a 15 fold increase over todays state. As a result, preparing energy infrastructure for the imminent impact on the electric grid is a pressing issue, particularly in California.

Even as EVs are beginning to have a major impact on the overall load of the grid, there is very little data around the load profiles of individuals with EVs. Figure 1a shows how different the average load profiles of households with and without EVs are, but depending on the EV car model, charging situation (whether a household has a smart charger), and house/household characteristics (i.e. house size, number of occupants, daily commutes...), these load profiles can vary significantly. Traditionally, load curves are modelled with a few fixed templates, but the load curves generated for households with EVs lack accuracy and supporting data. Therefore, due to the scarcity of load data for households with EVs, our aim in this research project is to accurately generate load profiles for households with EVs by using generative models.

Preprint. Work in progress.




(a) Average total household load with EV vs without EV.




(b) San Francisco net electric load on typical spring day characterizes a "Duck Curve", as measured by CaISO [2]

## 2 Related Work

Placing charging stations for EVs is a challenging problem. Tesla's strategy for placing charging stations largely depends on where their cars are being driven and popular driving routes. In other words, it is an optimization problem. While Tesla can estimate EV charging demand using proprietary data about their customers, governments and utilities also need to prepare charging stations and the grid for the upcoming disruption. Other researchers have approached EV load prediction with machine learning: B. Wang et al. implement a predictive scheduling framework for EV charging, [3] T. Zhang et al. use a Markov decision process to model user load,[4] and Y. Xiong et al. use clustering and then neural networks to model EV loads and perform load predictions. [5] Related, others have applied deep learning for solar power forecasting, and A. Gensler et al. achieved success combining AutoEncoders with LSTMs. [6]

Despite a few examples using machine learning for load prediction in the literature, the application area of generating realistic load curves using generative models is novel, so the bulk of the related work comes from other fields applying generative models to time series. Our first approach uses VAEs to generate time series; in "Variational Recurrent Auto-Encoders," [7] the authors showed that a recurrent VAE could be useful for initializing RNN weights for supervised learning. Our next experiment uses conditional VAEs to generate time series data and was partly inspired by "Semi-supervised learning with deep generative models" [8] and "Learning Structured Output Representation using Deep Conditional Generative Models." [9] The loss function that we use in our model is derived from the paper "Estimating the Mean and Variance of the Target Probability Distribution," [10] a method that quantifies uncertainty in neural networks and that is discussed in "High-Quality Prediction Intervals for Deep Learning." [11]

Others have used temporal autoencoding to improve the performance of generative models for time series datasets, as the authors in "Temporal Autoencoding Improves Generative Models of Time Series" [12] have done. Another area of research is applying GAN architectures for time series generation, as the authors of "Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs" [13] demonstrate that realistic generation of numeric time series is possible with recurrent GANs.

Many models can generate realistic time series for short sequences, like "WaveNet: A Generative Model for Raw Audio,"[14] the "Stochastic WaveNet"[15] and "WaveGlow," [16] but there have been fewer examples of generative models working well for low-dimensional real-valued series which is more relevant to our application. In "Hierarchical Implicit Models and Likelihood-Free Variational Inference,"[17] the authors implement a Bayesian GAN with a prior over its weights to generate sequence data for time series.

## 3 Task Definition

### 3.1 Dataset

In this project, we utilize the Pecan street dataset [18], containing electricity measurements for 1,115 different households over a five year period, with data updated each month. Our analysis is focused on

the 65 household subset that have electric vehicles. To get access to the data, we acquired Pecan Street university licenses and queried their PostgreSQL database. Our final dataset consists of continuous 15 minute interval load data from January 2015 to August 2018 for overall load and electric vehicle charging curves, along with corresponding metadata for each household.

In preprocessing, we interpolate missing data and filter out days with too many missing values. Days containing more than ten missing time steps in a row and those containing more than 25 percent missing overall were discarded. To increase the size of our dataset, as well as to introduce an element of randomness and to smooth out sharp edges, we created two additional files containing Loess smoothed curves across each month of data. When reading a sample $x$ from the data, we generate a random number $r \in [0, 1]$, and return

$$x = r \cdot x_{data} + (1 - r) \cdot x_{Loess}$$

We do this for both the EV curves and for the general load curves. A sample then consists of two size-96 vectors and a size-10 vector: the EV curve, the general use curve minus the EV curve, and metadata; for each day, we took the min temperature, the max temperature, the sum of precipitation, and the day of the week (one-hot encoded); these values are used as metadata, which we symbolize as model input $y$. We split our data into 80/10/10 train, validation, and test sets, containing 74356, 9294, and 9295 entries, respectively.

### 3.2 Objective

Our objective is to generate realistic EV charging curves for households without EVs, based on their general electric load curves. The idea is to be able to model how their load curves will change if they adapt an EV in the future. As building blocks, we learn to conditionally generate probabilistic general load curves and EV load curves. We do this on a per-day sample basis, without adding information about the particular household. This means that we rely on the load curve of a particular day to contain enough information about a household to infer when it would charge its EV. We tackle the objective of generating EV load curves for a particular household with two approaches, explained in the next section.

## 4 Approach

### 4.1 Proposed Methods

**Loss Definition** Both methods are trained by maximizing the evidence lower bound objective (ELBO) on their respective dataset. We use stochastic gradient descent to find $\theta, \phi$ that satisfy (1).

$$\arg\min_{\theta,\phi}(-ELBO(x;\theta,\phi)) \tag{1}$$

Where ELBO is given by (2). Our encoder and decoder $p_\theta$, $q_\phi$ are neural networks.

$$ELBO(x;\theta,\phi) = E_{q_\phi(z|x,y)}[\log p_\theta(x|z,y)] - D_{KL}(q_\phi(z|x,y)||p(z)) \tag{2}$$

The reconstruction loss component is given by the negative log-likelihood of the data given its latent representation and conditionals $-\log(p(x|z,y))$, assuming a heteroscedastic Gaussian distribution, where $\mu_\phi(z)$ and $\sigma^2_\phi(z)$ are parameterized by our neural network:

$$-\log(N(x|\mu_\phi(z|x,y), \sigma^2(z|x,y))) = \frac{1}{2}[\frac{(y-\mu_\phi(z|x,y))^2}{\sigma^2_\phi(z|x,y)} + \log(\sigma^2_\phi(z|x,y)) + \log(2\pi)] \tag{3}$$

In practice, we use a modified version of the above defined reconstruction loss, which no longer constitutes a valid log-probability:

$$L_{REC} = \frac{(y-\mu_\phi(z))^2}{\sigma^2_\phi(z)} + \log(\beta \cdot \sigma^2_\phi(z) + 1) \tag{4}$$

Apart from removing the constant terms, we add a term of 1 to the log, which we found empirically to help learn disentangled latent representations, as it decreases the importance of learning precise low-variance estimates of our noisy dataset, while still penalizing large variance estimates. Further,

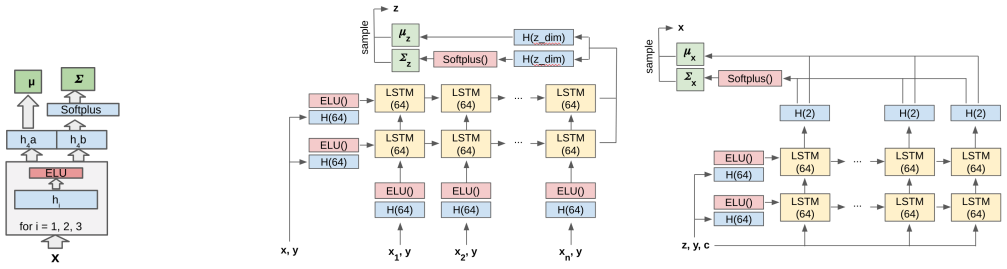
we introduce a weight $\beta$ that can be used increase the penalty for large variance estimates, which we found empirically to generate more detailed samples, particularly regarding load-peak estimation.

We separate the reconstruction loss in a term primarily dependent on $\mu_\phi(z)$ and $\sigma^2_\phi(z)$, which we will refer to simply as the scaled mean squared error (MSE) and variance term (VAR):

$$L_{MSE} = \frac{(y - \mu_\phi(z))^2}{\sigma^2_\phi(z)}$$

$$L_{VAR} = \log(\beta \cdot \sigma^2_\phi(z) + 1)$$

**Models** We compare two model architectures: A Feed-Forward Neural Network (FF-NN) and a LSTM Architecture, as shown in figures 2a and 2b. Further, we explore a smaller version of their architecture (FF-NN-s and LSTM-s), where we reduce both architecture to one hidden layer. Further, we explore a combination of a large encoder and a small decoder (FF-NN-s-dec and LSTM-s-dec). The FF-NN has hidden layers of dimension 128 and the LSTM of dimension 64.




(a) FF-NN.

(b) LSTM Encoder and Decoder.

Figure 2: CVAE Encoder and Decoder model architectures.

**Baseline Method 1: CVAE with supervised objective** In our first method, we train a conditional VAE, with two conditional terms. The first, categorical conditional $y$ corresponds to whether the load curve $x$, which we encode, includes EV load or not. We append it one-hot encoded to the input to the encoder and decoder. The second conditional $c$ is a continuous value, that corresponds to the EV load in $c \cdot 10kWh$. We multiply it with a two-dimensional one-hot vector, where position one refers to adding EV load to our encoded load curve, and position two to removing EV. This is a supervised objective, as we use our knowledge about the true EV load curve by changing the load curve $x_c$ over which we compute the reconstruction loss to the true load curve with added or removed EV load.

The training procedure involves retrieving batches from 2 datasets: a dataset with EV load separated into load without EV $x_0$ and load with EV $x_1$ and a dataset of loads without EV load $x_{none}$ For each batch run the CVAE to identity map all samples $x_0 \rightarrow_0$, $x_1 \rightarrow_1$ and $x_{none} \rightarrow_{none}$, by setting the conditional to zero. Additionally, we run the model over the batch with EV data to conditionally add/subtract EV load. Hereby, the encoded $x$ and decoded $x_c$ are $x_0 \rightarrow_1$ and $x_1 \rightarrow_0$, respectively.

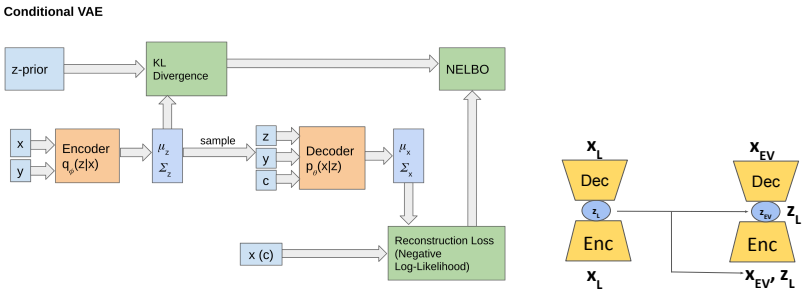
**Method 2: Dual CVAE** To relax the constraint of shared model parameters, we propose a dual architecture consisting of VAEs *V1* and *V2* (Figure 3b), whose goals are to learn representations of load curves sans EV, and EV curves, respectively. *V1* is trained solely on load curves without EV data, learning a latent representation for all households. *V2* takes a load curve as input, and first feeds it to *V1* to get a latent space representation, which is fed as additional input to the encoder. Intuitively, we generate EV curves conditioned on the corresponding general curve. By applying an architecture containing two models, we reduce parameter sharing, which helps generate better representations for each space. By conditioning *V2* on output from *V1*, we still learn the relationship between the two distributions.

In our dual CVAE, the objective for *V1* is identical to the baseline objective (1) for general load curves. For *V2*, we find $\theta_2, \phi_2$ that satisfy (4), which is the negative ELBO for sample $x$, conditioned on the latent vector output from the encoder of *V1*.

$$\arg\min_{\theta_2, \phi_2}(-ELBO(x; \theta_2, \phi_2 \mid y, p_{\theta_1}(x|y))) \tag{5}$$

Hereby, the conditional $y$ differs from the previous method, now representing our general metadata (weather and weekday). The conditional $c$ is the latent representation of the corresponding general load curve, encoded by VAE *V1*, which is only fed in to VAE *V2*.

We vary the encoder and decoder, using either feed-forward networks, or using LSTM models to better capture the time-series nature of our data. Furthermore, we alter the size of the architecture, removing one of the hidden layers in order to learn better general latent space representations.




(a) CVAE training setup. (b) Dual CVAE

Figure 3: Training setup for method 1 and method 2.

# 5 Results

## 5.1 Evaluation / Experimental procedures

We train our models on a 16-CPU Google Compute instance with 20GB memory, running each FF-NN for 20-40 epochs and each LSTM for 5-10 epochs on the 75k sample training dataset with batch size of 128. We used a latent space dimension of 5, learning rate annealing of 0.335 occurring every 5 epochs, and a default learning rate of 0.009. After training *V1* on general load curve, we found the small LSTM model with variance penalty of 1 to work well as the encoder and decoder, and used this architecture as our input to the EV load model *V2*, which we trained with identical hyperparameters, apart from a lower learning rate of 0.003.

To qualitatively evaluate the performance of our models, we created plots of usage and EV load curves conditioned on different $z$ vectors. By varying the magnitude of each element of the vector in isolation, we were able to determine whether each dimension was learning meaningful features (see 5.3).

## 5.2 Result Metrics

We evaluate our model using the negative ELBO (NELBO), KL divergence, and reconstruction loss. As discussed in Section 4.1, we use a modified version of the standard reconstruction loss, where we separate it into terms dependent on $\mu_\phi$ and $\sigma^2_\phi$, respectively. NELBO and KL divergence are computed in the usual manner, albeit with our modified $L_{Rec}$. We do not compare metrics directly between our baseline (Method 1) and final setup (Method 2), because it was computed over a different, smaller dataset with a different objective.

**Comparison of Metrics for Method 2:** The following loss metrics were obtained from training *V1* and *V2*, respectively.

As our dual VAE architecture is novel to this domain, we have no comparable models against which to evaluate directly. However, as the goal is to learn similarities between different household profiles, it would be interesting to apply clustering-based compression/sampling methods to see if similar relationships are learned.

Our metrics are tied to the data on which the model is run. For example, EV data typically has one or two large spikes during the day, with the load throughout the rest of the day being low to nonexistent. As such, it is easier for a good model to identify these spikes and drive down the error. On the other hand, general load curves have much more variation, so we may expect losses in general to be higher.

Table 1: Comparison of general load VAE model ( *V1*) variants and their performance on the validation set. If not otherwise stated, they were trained with a variance penalty (vp) of 10.

| Model | NELBO | KL | MSE | VAR | |
|---|---|---|---|---|---|
| | | | | $\beta = 1$ | $\beta = 10$ |
| FF-NN | 95.00 | 10.19 | 65.47 | 19.32 | 92.84 |
| FF-NN-s-dec | 94.12 | 10.37 | 62.99 | 20.76 | 96.88 |
| FF-NN-s | 98.39 | 9.99 | 68.96 | 19.43 | 93.85 |
| LSTM-s | 89.31 | 10.46 | 57.62 | 21.22 | 97.96 |
| LSTM-s (vp 1) | 74.02 | 7.29 | 33.61 | 33.11 | - |
| LSTM-s (vp 100) | 112.31 | 11.97 | 82.29 | 18.03 | - |
| LSTM-s (GM 100) | 87.38 | 10.01 | 56.15 | 21.17 | 99.53 |

Table 2: Comparison of conditional EV load VAE model ( *V2*) variants and their performance on the validation set. If not otherwise stated, they were trained with a variance penalty (vp) of 1.

| Model | NELBO | NIWAE (16) | KL | MSE | VAR $\beta = 1$ |
|---|---|---|---|---|---|
| FF-NN | 28.97 | 27.26 | 6.55 | 9.45 | 12.97 |
| FF-NN-s | 37.42 | 35.90 | 6.71 | 11.90 | 18.80 |
| FF-NN-s-dec | 36.75 | 35.51 | 6.54 | 11.61 | 18.59 |
| FF-NN-s-dec (vp 100) | 54.84 | 51.26 | 11.47 | 25.30 | 18.07 |
| FF-NN-s-dec (GM 100) | 33.77 | 32.99 | 5.04 | 10.85 | 17.87 |
| LSTM-s | 30.47 | - | 6.78 | 8.6 | 15.03 |

We found the loss metrics to be useful but not sufficiently explanatory of disentangled meaningful latent representations. Therefore we do not discuss the metrics in detail and focus on analyzing qualitative results.

### 5.3 Analysis - Latent Space

#### 5.3.1 Method 1

**Latent Space Variation** Figure 4 illustrates the variation of each dimension of our 5-dimensional latent vectors for our baseline GMVAE models. Each row shows the result of setting the corresponding $z_i = -2$ and 2, while keeping the other dimensions fixed to 0. Note that the LSTM model is run on hour-resolution data rather than the 15-minute resolution data. The LSTM GMVAE had difficulty with the higher resolution data, which we believe was a result of high noise. This was one reason we later added Loess smoothing.

We see the VAEs learn reasonably disentangled latent representations that approximate base load, peak time, curve shape, daytime/nighttime activity, and confidence in estimate. Generated random samples can be seen in Figure 13 (Appendices).

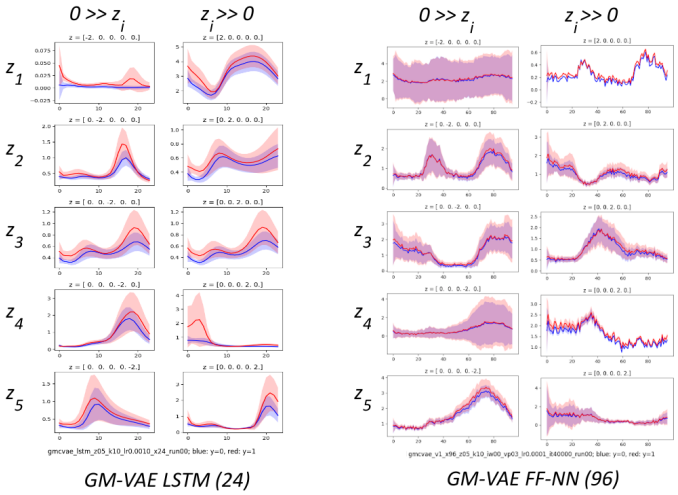
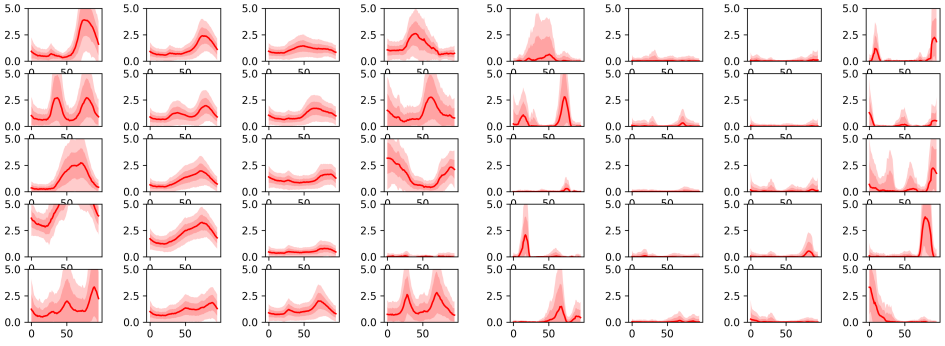



Figure 4: Sampled EV load curves from Method 1 FF-NN and LSTM when varying latent dimensions in the range $[-2, 2]$ while keeping the other dimensions fixed.

### 5.3.2 Method 2

**Latent Space Variation** As in section **5.3.1**, we examine the impact of varying different latent space variables on the generated EV load curve. The feed-forward neural network learns in most cases a meaningful dependence on the latent space, but is not always disentangled. The LSTM models often learn to rely on few of the already low-dimensional (5) latent space, but are better learning features that correspond to time, as can be seen in Figure 6, where varying this particular latent space variable captures the time of the charging peak.

When compared to the CVAE trained with both general load and EV load data (Figure 4), we see that *V1* generates less noisy load curves, with better confidence estimates. The EV load curves generated by *V2* (Figure 6) capture less variability, but still locate the primary EV charging times during the day. The same figures for an LSTM-s model can be seen in Appendices in figures 12a and 12b . Randomly generated samples can be found for both model architectures in Appendices, figures 14a, 14b and figures 15a, 15b.



(a) V1: General load curves. (b) V2: EV load curves.

Figure 5: Sampled General and EV load curves from V1 (FF-NN-s-dec) and V2 (FF-NN-s-dec) when varying latent dimensions in the range $[-3, 3]$ while keeping the other dimensions fixed.

To further analyze the latent space learned by our models, we link the outputs of each encoding to the house that provided the input, so that we can cluster households by latent representation and look for correlation with metadata. We use the validation set household data as inputs.

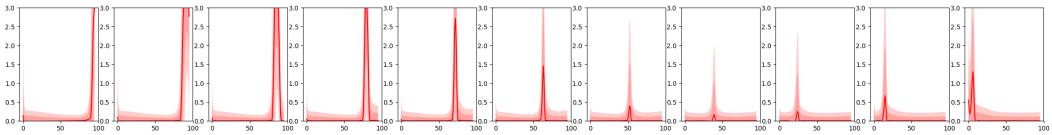
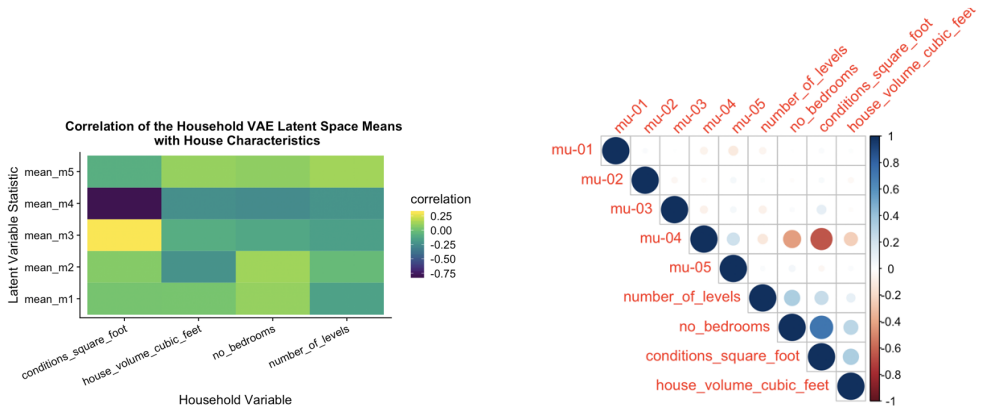


Figure 6: Sampled EV load curves from LSTM-s when varying latent dimension 3 in the range $[-3, 3]$ while keeping the other dimensions fixed.

We analyze the correlation between latent space and metadata for two different subsets of the metadata: (1) house size characteristics, and (2) EV charging preferences.

**Household Characteristics** We first analyze the correlation between latent space and house size. We pull hundreds of samples for every household and examine both the relationship between latent space values and household characteristics when averaged over each household, and the relationship between household characteristics and all pulled latent space values across all samples.




(a) Over household latent space mean

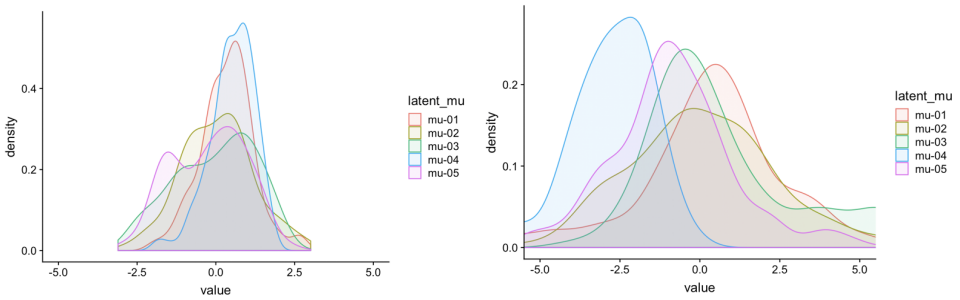
(b) Over all latent space samples

Figure 7: Pearson's Correlation of latent space variables and household metadata. See Table 7.4

Figure 7a shows the correlation between the means of every household's latent variables and our household metadata. We see that the latent space is very correlated with different house characteristics when you look at the latent spaces of different households' samples in aggregate. Figure 7b shows Pearson's Correlation plot relating the latent variable means of all of our validation samples with our household metadata.

Below is a case study comparing the latent spaces of two different households: one with the largest house in our dataset (4447 $ft^2$) and one with the smallest house in our dataset (1080 $ft^2$). These are density plots of the latent space means, and we have hundreds of latent space samples for each household.

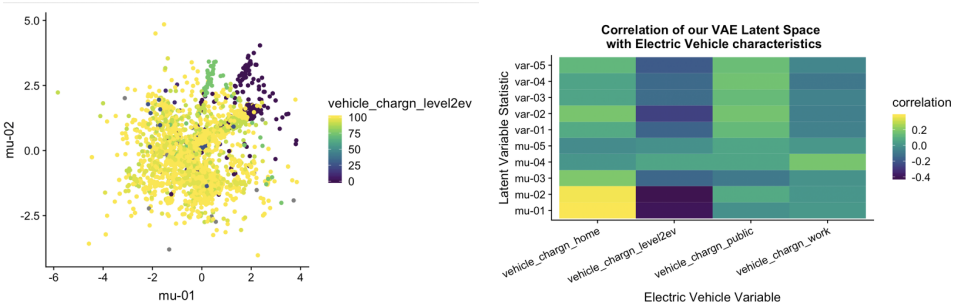
Figure 8a shows a density plot of the latent variables for the Household with the smallest house. Figure 8b shows a density plot of the latent variables for the Household with the largest house.




(a) smallest house (1080 sqft). (b) largest house (4447 sqft).

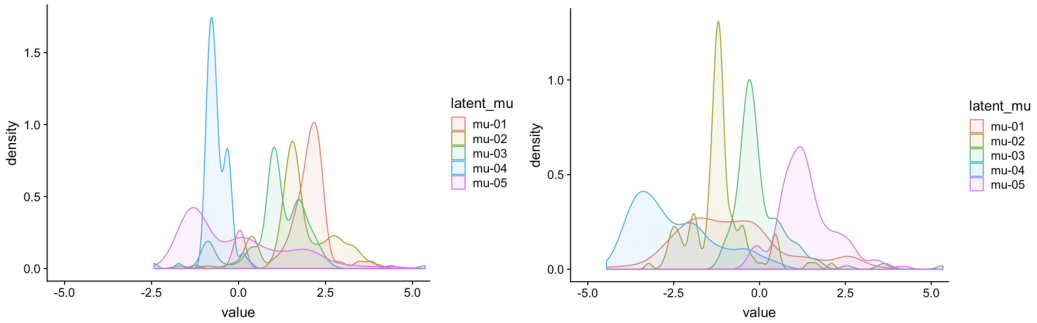
Figure 8: Density plots of two households' distributions over latent space.

**EV Preferences** Next, we analyze the relationship between the latent space, and the user's manner of charging in any given week. Figure 9a shows relatively distinct groupings in how mu-01 and mu-02 relate to the proportion of level2ev charging used. Figure 9b highlights that relatively strong correlations exist between mu-01, mu-02, and how the user charges their EV; in particular, whether they have slow or fast charging at home. Between Figure 10a and Figure 10b, we see general differences in the latent space distributions according to where the user charges their EV.




(a) Scatterplot of EV Latent Space with Percentage of Charging that is Level 2 (b) Correlation of our VAE Latent Space with Electric Vehicle characteristics

Figure 9: Relationship between EV charging characteristics and Latent Space. See Table 7.4




(a) Latent distributions for household that claims to charge 100% at home (b) Latent distributions for household that claims to charge 0% at home

Figure 10: Density plots of two households' distributions over latent space.

### 5.4 Analysis - Conditional EV Curve Generation

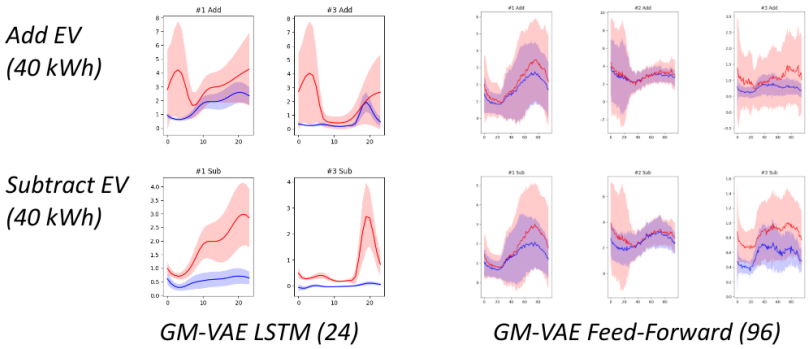



Figure 11: Method 1: Generated load curves corresponding to adding or subtracting an EV Load of approx $40\ kWh$. Results shown for a Gaussian mixture VAE LSTM, modelling hourly load and a Gaussian mixture VAE FF-NN with 15 minute resolution data.

Our baseline method 1 performs does poorly when required to add EV measurements to a base curve. In particular, we found feed-forward networks to be more adept at adding and subtracting from general and EV curves, but they had more difficulty learning reasonable representations of load curves. LSTM networks were better at learning reasonable shape of the added EV curve, but were inflexible when adding and subtracting. But both models do not go beyond adding an average EV-load curve atop the existing load curve, failing to learn any dependence on the general load curve to determine the EV load shape for this particular sample.

Results from our method 2 show no learned conditional dependence on the general load curve latent representation. Varying the conditional latent dimensions leads to no observable change in the generated EV load curve, as seen in figure 16a. Similarly, we failed to observe a dependence on weekdays.

## 6 Conclusion

In this paper, we proved the efficacy of conditional VAEs for generating reasonable load curves. Furthermore, we demonstrated the benefits of using a dual CVAE architecture to learn the general load distribution before training on EV load data, as it leads to smoother and more sensible samples (Figure 5a). We showed that the latent spaces learned correlate with relevant metadata such as house size and access to at-home charging, despite the model not having had access to this metadata in training. This in particular may prove useful for modeling load curves in practice, as load data is far more easily accessible than household metadata.

Despite learning good latent representations for EV curves and for general use curves using our dual CVAE architecture, the model did not learn a dependence between the two latent distributions. As our analysis of the latent space and the correlations to a households overall characteristic showed significant results, we would recommend a future approach to use a per-household and not a per-day sampling method in order to achieve this objective. This was, however, not possible with our dataset, as 65 households is unlikely enough for this objective. Nevertheless, the learned latent representations can be useful for a downstream task such as characterizing households by their loads, predicting future loads, given a households historic latent representations or building a simulator for a reinforcement learning algorithm.

## References


[1] T. Bunsen, P. Cazzola, M. Gorner, L. Paoli, S. Scheffer, R. Schuitmaker, J. Tattini, and J. Teter, "Global ev outlook 2018," *International Energy Agency*, 2018.

[2] C. ISO, "Demand response and energy efficiency roadmap: Maximizing preferred resources.," 2013.

[3] B. Wang, Y. Wang, H. Nazaripouya, C. Qiu, C.-C. Chu, and R. Gadh, "Predictive scheduling framework for electric vehicles considering uncertainties of user behaviors," *IEEE Internet of Things Journal*, pp. 1–1, 2016. DOI: 10.1109/jiot.2016.2617314.

[4] T. Zhang, W. Chen, Z. Han, and Z. Cao, "Charging scheduling of electric vehicles with local renewable energy under uncertain electric vehicle arrival and grid power price," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2600–2612, Jul. 2014, ISSN: 0018-9545. DOI: 10.1109/TVT.2013.2295591.

[5] Y. Xiong, B. Wang, C.-C. Chu, and R. Gadh, "Electric Vehicle Driver Clustering using Statistical Model and Machine Learning," *arXiv e-prints*, Feb. 2018. arXiv: 1802.04193.

[6] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting — an approach using autoencoder and lstm neural networks," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2016, pp. 002 858–002 865. DOI: 10.1109/SMC.2016.7844673.

[7] O. Fabius and J. R. van Amersfoort, "Variational Recurrent Auto-Encoders," *arXiv e-prints*, Dec. 2014. arXiv: 1412.6581 [stat.ML].

[8] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-Supervised Learning with Deep Generative Models," *arXiv e-prints*, Jun. 2014. arXiv: 1406.5298.

[9] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., 2015, pp. 3483–3491. [Online]. Available: http://papers.nips.cc/paper/5775-learning-structured-output-representation-using-deep-conditional-generative-models.pdf.

[10] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 1, Jun. 1994, 55–60 vol.1. DOI: 10.1109/ICNN.1994.374138.

[11] T. Pearce, M. Zaki, A. Brintrup, and A. Neely, "High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach," *arXiv e-prints*, Feb. 2018. arXiv: 1802.07167 [stat.ML].

[12] C. Häusler, A. Susemihl, M. P. Nawrot, and M. Opper, "Temporal Autoencoding Improves Generative Models of Time Series," *arXiv e-prints*, Sep. 2013. arXiv: 1309.3103 [stat.ML].

[13] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs," *arXiv e-prints*, Jun. 2017. arXiv: 1706.02633 [stat.ML].

[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arXiv e-prints*, Sep. 2016. arXiv: 1609.03499 [cs.SD].

[15] G. Lai, B. Li, G. Zheng, and Y. Yang, "Stochastic WaveNet: A Generative Latent Variable Model for Sequential Data," *arXiv e-prints*, Jun. 2018. arXiv: 1806.06116.

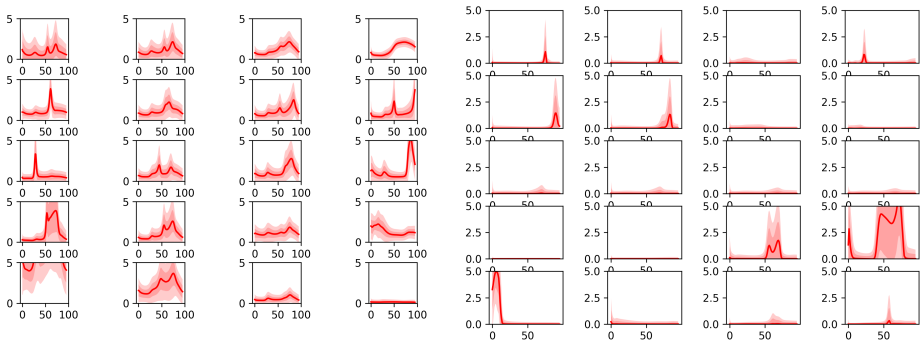
[16] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A Flow-based Generative Network for Speech Synthesis," *arXiv e-prints*, Oct. 2018. arXiv: 1811.00002 [cs.SD].

[17] D. Tran, R. Ranganath, and D. M. Blei, "Hierarchical Implicit Models and Likelihood-Free Variational Inference," *arXiv e-prints*, Feb. 2017. arXiv: 1702.08896 [stat.ML].

[18] C. A. Smith, "The pecan street project: Developing the electric utility system of the future," *University of Texas, Austin, TX*, 2009.

# 7 Appendices

## 7.1 Latent Space Variation



(a) V1: General load curves.

(b) V2: EV load curves.

Figure 12: Sampled General and EV load curves from V1 (LSTM-s) and V2 (LSTM-s) when varying latent dimensions in the range $[-3, 3]$ while keeping the other dimensions fixed.

## 7.2 Generated Random Samples

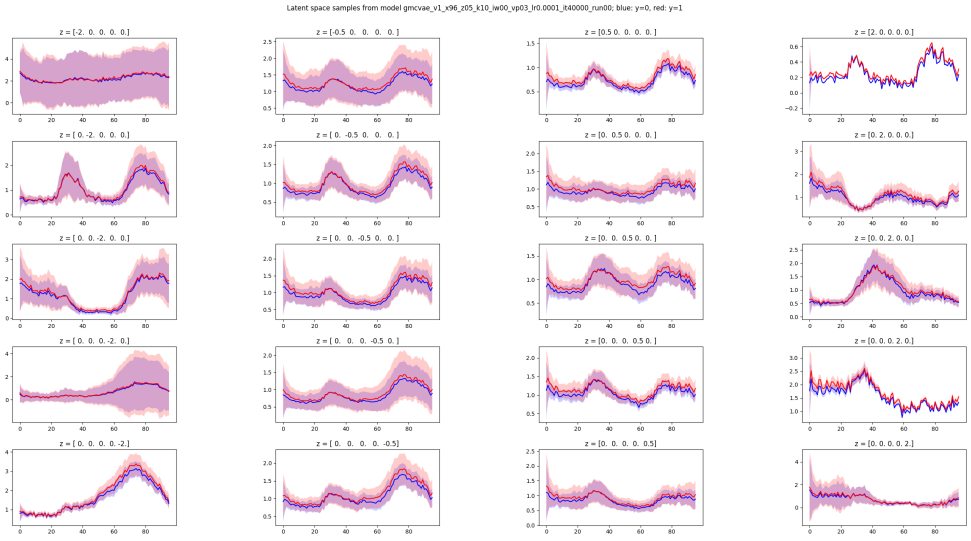
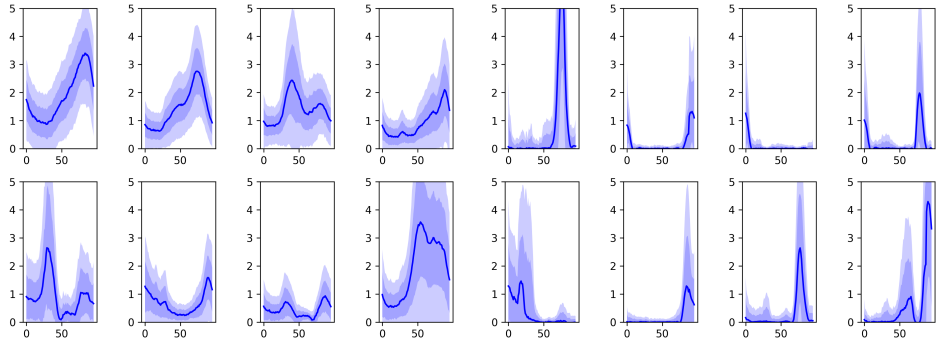
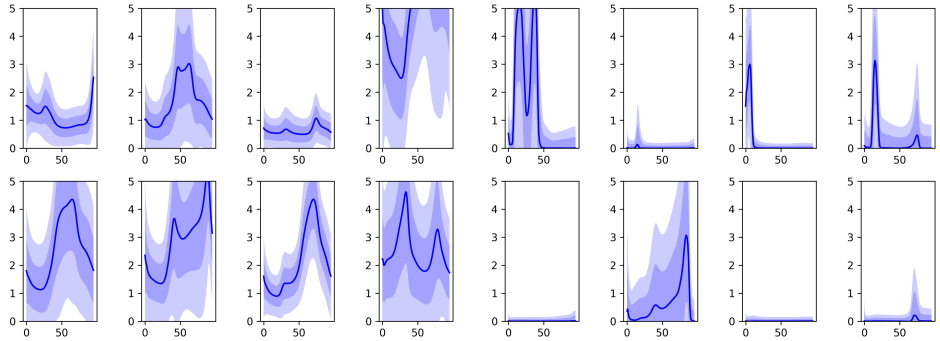



Figure 13: Method 1: Random Samples from GMVAE over 15minute resolution data. Blue: conditional set to no-EV. Red: conditional set to EV-load.



(a) General load curves from V1. (b) EV load curves from V2.

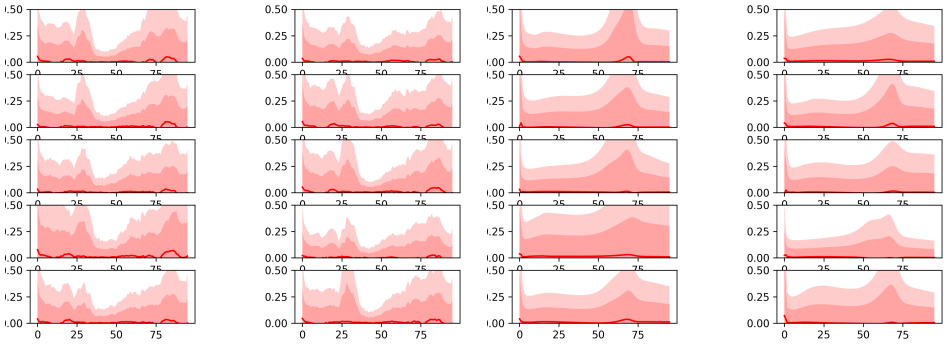
Figure 14: Method 2: Randomly generated samples V1 and V2 (FF-NN-s-dec).



(a) General load curves from V1. (b) EV load curves from V2.

Figure 15: Method 2: Randomly generated samples V1 and V2 (LSTM-s).

### 7.3 Conditional EV curve generation



(a) FF-NN-s-dec (b) LSTM-s

Figure 16: Method 2: Conditionally generated EV load curves when varying one dimension of the V1 latent space on which the V2 Dual model is conditioned on.

### 7.4 Information about Data

| Household Characteristic | Definition |
|---|---|
| number_of_levels | Number of levels in the house |
| no_bedrooms | Number of bedrooms |
| conditions_square_foot | Area of house in $ft^2$ |
| house_volume_cubic_feet | Volume of house in cubic feet |

| EV Preference | Definition |
|---|---|
| vehicle_chargn_home | Percentage of EV charging at home in average week |
| vehicle_chargn_level2ev | Percentage of EV charging that is Level 2 charging in average week |
| vehicle_chargn_public | Percentage of EV charging in public charging stations in average week |
| vehicle_chargn_work | Percentage of charging at work in average week |

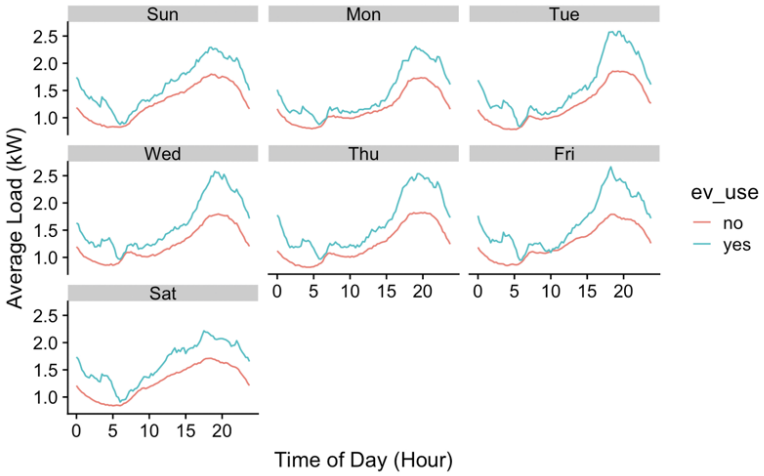



Figure 17: Average load curves for households with and without EVs for every day of the week