Temporal Attention Bottleneck is informative? Interpretability through Disentangled Generative Representations for Time Series Disaggregation

Khalid Oublal [*1,2] **Saïd Ladjal** [1] **François Roueff** [1] **David Benhaiem** [2] **Emmanuelle le-borgne** [2]

Abstract

Generative models have garnered significant attention for their ability to address the challenge of source separation in disaggregation tasks. Energy Disaggregation holds promise for promoting energy conservation by allowing homeowners to gain comprehensive insights into their energy consumption solely through the interpretation of aggregated load curves. Nevertheless, the model's ability to generalize and its interpretability remain two major challenges. To tackle these challenges, we deploy a generative model called TAB-VAE (Temporal Attention Bottleneck for Variational Auto-encoder), based on hierarchical architecture, addresses signature variability, and provides a robust, interpretable separation through the design of its informative representation of latent space. Our implementation and evaluation guidelines are available at https://github.com/oublalkhalid/TAB-VAE.

1. Introduction

Rising interest in reducing carbon footprints through the user's energy activity poses new challenges to traditional solutions. In fact, most households rely on their monthly bills from previous months to adjust their energy use for the following month. Therefore, Energy Disaggregation is a non-intrusive way of monitoring the energy consumption of individual appliances from an aggregated load profile (a mixed signal). In recent years, deep models have been widely used for Energy Disaggregation due to their ability to learn complex patterns in data. Nevertheless, these approaches often suffer from concerns related to interpretability, making it difficult to understand the decisions made, and on the other hand, they are in general less generalizable and unrobust. To address these concerns, several approaches have been proposed, such as the use of convolutional neural networks (CNNs) to extract features from the consumed power shapes, proposed by (Ciancetta et al., 2021). Although this approach has shown promising results on the UK-DALE dataset (Kelly & Knottenbelt, 2015), it has a generalization problem. In previous works, (Chen et al., 2018a) introduced a sequence-in-sequence (S2S) approach combining CNNs and LSTM for Energy Disaggregation, while (Yang et al., 2021) proposed a novel RNN-based method called S2P, which utilizes GRUs and attention mechanisms for improved performance. However, the lack of interpretability and generalizability in these deep-learning approaches remains a significant concern. Thus, our paper focuses on addressing these challenges by investigating the effectiveness of the Generative Temporal Attention Bottleneck for separation generalizability and effectively learn an interpretable representation of the latent space.

2. Problem Statement and Motivation

Let $\mathbf{X}^{(t)} := \mathbf{X}_{t:t+\tau} \in \mathbb{R}^{C\times\tau}$ be a sequence of the aggregate measured power noise for the whole household for the range time $t : t+\tau$, and with $C = 3$ (corresponding to the active, reactive and apparent power)[1]. We note $X_t \in \mathbb{R}$ the active power, which is written as the sum of the contributions of each device $y_{t,m}$, $m = 1, \ldots, M$, and a residual noise ξ_t:

$$X_t = \sum_{m=1}^{M} y_{t,m} + \xi_t \tag{1}$$

The index m refers to the m-th electrical device among the M available. The problem is to deduce, from a sequence

[1]Department of Computer Science, Institute Polytechnique de Paris, Telecom Paris, 19 Pl. Marguerite Perey, 91120 Palaiseau, France. [2]OneTech, TotalEnergies SE Tour Coupole - 2 place Jean Millier 92078 Paris la Défense. Correspondence to: Khalid Oublal <khalid.oublal@polytechnique.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1]The electric power is composed of three types: active, reactive, and apparent. The active power is the quantity of electrical energy actually converted into useful work. The reactive power is the amount of electrical energy temporarily stored and temporarily exchanged between the power source and the electrical device, without the electrical device, producing useful work. The apparent power is the vectorial sum of the active and reactive power.

$\mathbf{X}^{(t)}$ of length τ, the corresponding components $Y^{(t)} := y_{1:M}^{(t)} := y_{t:t+\tau,1:M}$.

We note $\mathcal{D} = \{\mathbf{X}^{(t)}, Y^{(t)}\}_{t=1}^{N}$ the training set, where $\mathbf{X}^{(t)} \in \mathbb{R}^{C\times\tau}$ and $Y^{(t)} \in \mathbb{R}^{M\times\tau}$. $\mathbf{X}^{(t)}$ represent a set of samples from an unknown distribution. Variational Auto-Encoder (VAE) aims at inferring this distribution with a parametric model with a latent (unobserved) variable. We define this latent variable as a $Z^{(t)}$ variable of dimension $(M+1)\times d_z$ representing a multivariate sequence $\mathbf{X}^{(t)}$. In others for clarity and consistency with existing literature, In this paper, we use two notations $Z^{(t)}$ and Z (respectively $Z_l^{(t)}$ and Z_l) interchangeably to denote the same underlying quantity $Z^{(t)}$.

Inference in the generative model involves computing the marginal likelihood $p(\mathbf{X}^{(t)})$ by integrating out the latent variables: $p(\mathbf{X}^{(t)}) = \int p(\mathbf{X}^{(t)}, Z^{(t)})dZ^{(t)}$. However, this integration is often intractable. To address this, the Evidence Lower Bound (ELBO) is introduced by (Kingma & Welling, 2013) as an objective function, which can be optimized efficiently using stochastic gradient descent. The ELBO provides a lower bound on the marginal likelihood and plays a key role in variational inference with continuous latent variables.

$$\log p(\mathbf{X}^{(t)}) \geq \mathbb{E}_{q(Z^{(t)}|\mathbf{X}^{(t)})}\left[\log p(\mathbf{X}^{(t)}|Z^{(t)})\right] - \mathrm{KL}(q(Z^{(t)}|\mathbf{X}^{(t)}) \parallel p(Z^{(t)})) \quad (2)$$

where θ, ϕ parameterize $p(\mathbf{X}^{(t)}, Z^{(t)}); \theta)$ (denote by $p_\theta(.)$) and $q(Z^{(t)})|\mathbf{X}^{(t)}; \phi)$ (denote by $p_\phi(.)$) respectively.

Recently (Vahdat & Kautz, 2020) introduced NVAE, an extended VAE with a hierarchical latent variable model for structured representations. It also involves a sum over the layers, computing the expected KL divergence between the posterior distribution $q_\phi(Z_{<l}^{(t)}|\mathbf{X}^{(t)})$ and the prior distribution $p_\theta(Z_l^{(t)}|Z_{<l}^{(t)})$ to assess the alignment of inferred latent variables with the prior. However, it lacks the ability to capture the temporal context in time series data. To address this limitation, we propose Temporal Attention Bottleneck mechanism, introducing attention to capture temporal dependencies and improve inference contextual learning.

Main Contribution: Our work enhances the flexibility of the prior distribution $p(Z)$ and posterior distribution $q_\phi(Z^{(t)}|\mathbf{X}^{(t)})$ by introducing informative representations for the conditional distributions $p_\theta(Z_l^{(t)}|Z_{<l}^{(t)})$ and $q_\phi(Z_l^{(t)}|\mathbf{X}^{(t)}, Z_{<l}^{(t)})$. We achieve this through a hierarchical structure of densely connected stochastic layers, improving the model's capacity to capture complex data dependencies. Figure 2 illustrates our proposed model. Additionally, we thoroughly evaluate different prior distributions (Gaussian vs. Spherical) using various datasets and metrics to assess their impact on model performance.

3. Proposed Methods

Our approach for Energy Disaggregation aims to identify and accurately separate the contributions of the different appliances $y_m^{(t)}$ for $m = 1 : M$ in a given aggregated power sequence $X_{t:t+\tau}$. Let (f_ϕ, f_θ) be an encoder/decoder pair. This means that, given the latent code $Z^{(t)}$, $\mathbf{X}^{(t)}$ follows a law parameterized by $f_\theta(Z^{(t)})$. On its side, the latent code $Z^{(t)}$, given $\mathbf{X}^{(t)}$, follows a law parameterized by $f_\phi(\mathbf{X}^{(t)})$. The latent code can be factorized as: $z = z_{1:M+1}$ where $z_m \in \mathbb{R}^{d_z}$ represents the latent code of device $m = 1 : M$, while $z_{M+1} \in \mathbb{R}^{d_z}$ represents the latent code of remnant ξ. The dimension d_z corresponds to the dimension required to encode the signal signature of each device along a sequence of size τ.

3.1. Loss function, Proxy for Energy Disaggregated

The approximation of the conditional distribution of Z is given by a Gaussian variational distribution $q_\phi(Z^{(t)}|\mathbf{X}^{(t)}) = \mathcal{N}(Z^{(t)}; \mu(\mathbf{X}^{(t)}, \phi), \sigma^2(X^{(t)}, \phi))$, where $\mu(\mathbf{X}^{(t)}, \phi)$ and $\sigma^2(\mathbf{X}^{(t)}, \phi)$ are the outputs of the residual unit, which parameterize the mean and variance of the distribution, respectively. In our approach, the KL term is identical to (Vahdat & Kautz, 2020): $\mathrm{KL}(q_\phi(\mathbf{z}^{(t)}|\mathbf{X}^{(t)})||p(\mathbf{z}^{(t)}))$, with p denoting the a priori distribution of $Z^{(t)}$ taken as a centered and normalized Gaussian, which leads to the $\mathcal{L}_{KL}$ term:

$$\begin{aligned}\mathcal{L}_{KL} = \frac{1}{2}\sum_{j=1}^{J}(&\log\sigma_j^2(\mathbf{X}^{(t)}, \phi) \\ &- \mu_j^2(\mathbf{X}^{(t)}, \phi) - \sigma_j^2(\mathbf{X}^{(t)}, \phi) + 1) \\ &+ \sum_{l=2}^{L}\mathbb{E}_{q(Z_l|\mathbf{X}^{(t)})}\left[\mathrm{KL}(q(Z_l|\mathbf{X}^{(t)}, Z_{<l})||p(Z_l|Z_{<l}))\right]\end{aligned} \quad (3)$$

where $J = (M+1)\, d_z$. In contrast, the reconstruction term is slightly modified and can be defined as follows:

$$\mathcal{L}_{rec} = \frac{1}{\tau}\sum_{t}^{t+\tau}\sum_{m=1}^{M}||y_{t,m} - \hat{y}_{t,m}||^2 \quad (4)$$

where $\hat{y}_{t,m}$ denotes the power predicted at time t by the m-th output of the f_θ decoder applied to a $Z^{(t)}$ simulated under the ϕ parameter.

3.2. Temporal Attention Bottleneck

Unlike NVAE (Vahdat & Kautz, 2020) for which the latent space Z is level-structured locally, in this work, we introduce *Temporal Attention Bottleneck* (TAB), which enabling the model to establish strong couplings, as depicted in Figure 2 and motivated in Section 2. The core problem we

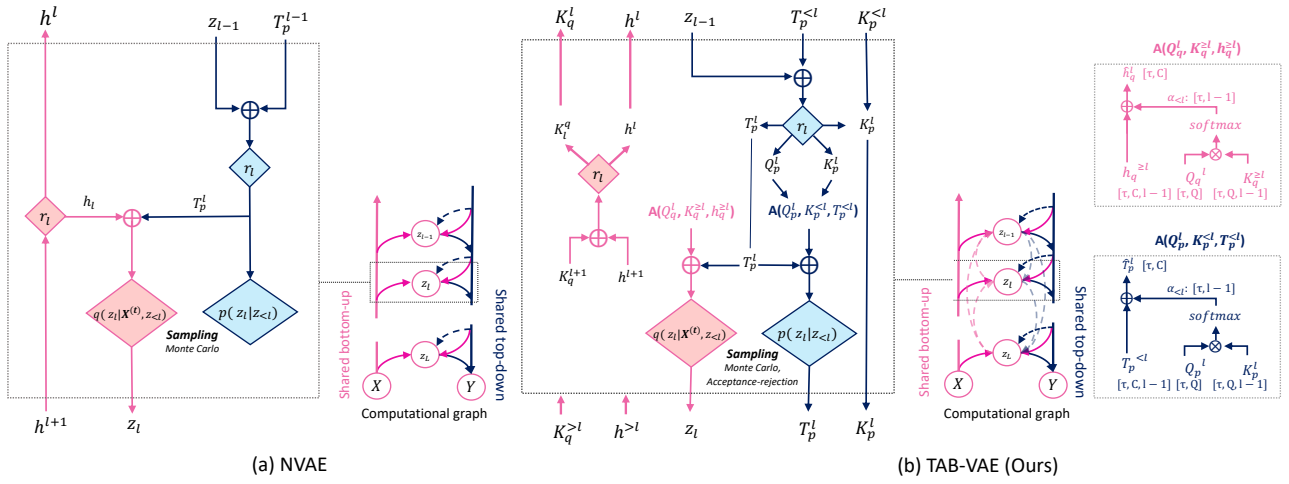

Figure 2. Left, Figure 2-a, NVAE involves connecting each layer only to adjacent layers and using T_p^{l-1} to carry latent information from earlier layers. **Inference** computes $q_\phi(Z_l^{(t)}|x, Z_{<l}^{(t)})$ with T_q^l, while **Generation path** $p_\theta(Z_l^{(t)}|Z^{(t)}< l)$ with T_p^l (T_p^0 is learnable parameter initialized by zero). The operator $\oplus$ combines information from two branches in the network, r_l and r_l represenet residual layers at level l respectively in encoder and decoder. In Figure 2-b, layers connect with all below (above) during bottom-up (top-down) passes, using attention modules to capture temporal dependencies and improve generative results.

aim to address is to construct a feature $\hat{T}^l$ that effectively captures the most informative features of time series from a given sequence $T^{<l} = \{T^i\}_{i=1}^l$ of l contexts for a given task. Both $\hat{T}^l$ and T^l are features with the same dimensionality: $\hat{T}^l \in \mathbb{R}^{\tau \times C}$ and $T^i \in \mathbb{R}^{\tau \times C}$. In our framework, we employ Temporal Attention to construct either the prior or posterior beliefs of a variational layers, which enables us to handle long context sequences with large dimensions τ effectively. The construction of $\hat{T}^l$ relies on a query feature $\mathbf{Q}^l \in \mathbb{R}^{\tau \times Q}$ of dimensionality Q with $Q \ll C$, and the corresponding context T^l is represented by a key feature $\mathbf{K}^l \in \mathbb{R}^{\tau \times Q}$. Importantly, $\hat{T}^l(t)$ of time step i in sequence τ depends solely on the time instances in $T^{<l}$.

$$\hat{T}^l(t) = \sum_{i<l} \alpha_{i \to l}(t) \cdot T^l(t), \tag{3}$$

$$\alpha_{i \to l}(t) = \frac{\exp(Q_l^\intercal(t) \cdot \mathbf{K}^l(t))}{\sum_{i<l} \exp(Q_l^\intercal(t) \cdot \mathbf{K}^l(t))}.$$

In words, feature $\mathbf{Q}^l(t) \in \mathbb{R}^Q$ queries the Temporal significance of feature $T^l(t) \in \mathbb{R}^C$, represented by $\mathbf{K}^l(t) \in \mathbb{R}^Q$, to form $\hat{T}^l(t) \in \mathbb{R}^C$. $\alpha_{i \to l}(t) \in \mathbb{R}$ is the resulting relevance metric of the i-th term, with $i < l$, at time step t. The overall procedure is denoted as $\hat{T} = \mathbf{A}(T^{<l}, \mathbf{Q}^l, \mathbf{K}^{<l})$, and is illustrated in Figure 2-b.

Generative Model p_θ. As shown in Figure 2-a, the conditioning factor of the prior distribution at variational layer l is represented by context feature $T_p^l \in \mathbb{R}^{\tau \times C}$. A convolution is applied on T_p^l to obtain parameters θ defining the prior. $\mathbf{T}_p^l$ is a non-linear transformation of the immediately previous latent information $Z_l^{(t)}$ and prior context T_p^l containing latent information from distant layers $Z_{<l}^l$, such that $T_p^l = \mathbf{T}_p^l(Z_l^{(t)} \oplus T_p^l)$. $\mathbf{T}_p^l(\cdot)$ is a transformation operation, typically implemented as a cascade of residual cells and corresponds to the blue residual module in Figure 2-a. $Z_l^{(t)}$ and T_p^l are passed in from the previous layer. Because of the architecture's locality, the influence of $Z_l^{(t)}$ could potentially overshadow the signal coming from T_p^l. To prevent this, we adopt direct connections between each pair of stochastic layers, as shown in Figure 2-b. That is, variational layer l has direct access to the prior temporal context of all previous layers $T_p^{<l}$ accompanied by keys $\mathbf{K}_p^{<l}$. This means each variational layer can actively determine the most important latent contexts when evaluating its prior beliefs. During training, the temporal context T_p, $\mathbf{Q}_p$, and $\mathbf{K}_p$ are jointly learned:

$$[T_p^l, \mathbf{Q}_p^l, \mathbf{K}_p^l] \leftarrow \mathbf{T}_p^l(Z_l^{(t)} \oplus T_p^l) \text{ for } l = L, L-1, ..., 1.$$

We initially let variational layer l rely on nearby dependencies captured by T_p^l. During training, the prior is progressively updated with the holistic context $\hat{T}_p^l$ via a residual connection:

$$\hat{T}_p^l \leftarrow \mathbf{A}(T_p^{<l}, \mathbf{Q}_p^l, \mathbf{K}_p^{<l})$$

$$\hat{T}_p^l \leftarrow T_p^l + \eta_p^l \hat{T}_p^l \text{ for } l = L, L-1, ..., 1.$$

where $\eta_p^l \in \mathbb{R}$ is a learnable scalar parameter initialized by zero, $T_p^{<l} = \{T_p^i\}_{i=1}^l$ with $T_p^i \in \mathbb{R}^{\tau \times C}$, $\mathbf{Q}_p^l \in \mathbb{R}^{\tau \times Q}$, $\mathbf{K}_p^{<l} = \{\mathbf{K}_p^i\}_{i=1}^l$ with $\mathbf{K}_p^i \in \mathbb{R}^{\times Q}$, and $Q \ll C$.

Infrence Model q_ϕ. As shown in Figure 2, the conditioning context T_q^l of the posterior distribution results from combining deterministic factor h^l and stochastic factor T_p^l provided by the decoder: $T_q^l = h^l \oplus T_p^l$. To improve inference, we let layer l's encoder use both its own h^l and all subsequent hidden representations $h^{\geq l}$, as shown in Figure 2. As in the generative model, the bottom-up path is extended to emit low-dimensional key features $\mathbf{K}_q^l$, which represent hidden features h^l:

$$[h^l, \mathbf{K}_q^l] \leftarrow \mathbf{T}_q^l(h_{l+1} \oplus \mathbf{K}_q^{l+1}) \text{ for } l = L, L-1, ..., 1.$$

Prior works (Vahdat & Kautz, 2020) have sought to mitigate against exploding Kullback-Leibler divergence (DKL) in Equation 2 by using parametric coordination between the prior and posterior distributions. Motivated by this insight, we seek to establish further communication between them. We accomplish this by allowing the generative model to choose the most explanatory features in $h^{\geq l}$ by generating the query feature $\mathbf{Q}_q^l$. Finally, the holistic conditioning factor for the posterior is:

$$\hat{T}_q^l \leftarrow \mathbf{A}(h^{\geq l}, \mathbf{Q}_q^l, \mathbf{K}_q^{\geq l}) \text{ for } l = L, L-1, ..., 1. \quad (5)$$

We adopt the Gaussian residual parametrization between the prior and the posterior proposed by (Vahdat & Kautz, 2020). The prior is given by:

$$p(Z_l^{(t)}|Z_{<l}) = \mathcal{N}(\mu(T_p^l, \theta), \sigma(T_p^l, \theta). \quad (6)$$

The posterior is then given by:

$$q(Z_l^{(t)}|\mathbf{X}^{(t)}, Z_{<l}^{(t)}) = \mathcal{N}(\mu(T_p^l, \theta) + \mathbf{\Delta}\mu(\hat{T}_q^l, \phi), \quad (7)$$
$$\sigma(T_p^l, \theta) \cdot \mathbf{\Delta}\sigma(\hat{T}_q^l, \phi))$$

where the sum $(+)$ and product $(\cdot)$ are pointwise, and T_q^l is defined in Eq.5. $\mu(\cdot)$, $\sigma(\cdot)$, $\mathbf{\Delta}\mu(\cdot)$, and $\mathbf{\Delta}\sigma(\cdot)$ are transformations implemented as convolutions layers. The inference procedure is also described in detail in Algorithm-1 Applendix.A.3. For $\mathcal{L}_{KL}$ in 3, the last terme is approximated by: $0.5\left(\frac{\mathbf{\Delta}\mu_1^2}{\sigma_l^2} + \mathbf{\Delta}\sigma_l^2 - \log \mathbf{\Delta}\sigma_l^2 - 1\right)$.

Impact of Gaussian Distribution Prior. In low dimensions, the Gaussian prior causes clustering around the origin, making it problematic for multiple clusters. An ideal prior should increase variance without biasing the mean. A uniform prior satisfies this but isn't well-defined on the hyperplane. In high dimensions, the Gaussian approximates a uniform distribution on a hypersphere's surface due to the "soap bubble effect." Comparing it with a naturally defined hypersphere posterior is motivated by this and concerns regarding the curse of dimensionality from the L2 norm.

von Mises-Fisher TAB (vMF-TAB). The von Mises-Fisher distribution is a distribution on the $(d-1)$-dimensional sphere in $\mathbb{R}^d$. The vMF distribution is defined by a direction vector μ with $||\mu|| = 1$ and a concentration parameter $\kappa \geq 0$. The PDF of the vMF distribution for the d-dimensional unit vector $\mathbf{X}^{(t)}$ is defined as:

$$f_d(\mathbf{X}^{(t)}; \mu, \kappa) = C_d(\kappa) \exp(\kappa \mu^T \mathbf{X}^{(t)})$$

where,

$$C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}$$

and I_v represents the modified Bessel function of the first kind at order v. In Table 3, we display the results obtained using TAB-VAE with the vMF distribution. Although there is only a minor difference in performance, this calls for a more comprehensive investigation in this direction. All theoretical proofs to compute KL for such distribution are given in (Davidson et al., 2018), an update has been developed in Appendix.A.4 to support our TAB-VAE.

3.3. Explicability underlying latent space structuring

An interpretable representation of learning is obtained when the latent space is factorized and the multidimensional components are statistically independent, which is a complex task in the context of information theory for generative models. A variety of methods have been proposed to solve this problem, such as β-TCVAE (Chen et al., 2018c). The most commonly used method is derived from the information theory known as *Total Correlation*, which introduces the TC penalty that is defined by the divergence $\mathrm{KL}(p_\phi(Z)||\prod_j p_\phi(z_j))$. Nevertheless, estimating this divergence is both expensive and difficult to perform.

Estimation of TC. To avoid costly *TC* estimation and guarantee time-series robustness, we try to apply this penalty using a discriminator across Z. It has been previously used as a disentangling metric for image generation (Chen et al., 2018b). In our case, we use it as a loss function. For its training, the latent variables of half the batch are randomly permuted, creating positive z_{perm} *(i.e all components are independent)*, and the other half is left untouched, corresponding to negative case *(i.e components are correlated)*. A D_ψ discriminator is used to replace the penalty, denoted TC in the following, by optimizing the performance of a discriminator between the distribution of the latent variable and a permuted of it. The D_ψ discriminator and the model are trained simultaneously.

$$\mathcal{L}_{TC} = \mathbb{E}[\log(D_\psi(z_{\text{permuted}}))] + \mathbb{E}[\log(1 - D_\psi(Z))] \quad (8)$$

Organizing and Alignment of Z by Masking. Our aim is to match the z_j component to the j device using a masking policy during training. Instead of giving as input a normal sequence $\mathbf{X}^{(t)}$, we give as input a sequence $y_j^{(t)}$ corresponding to a device j. The only evaluated output is corresponding

to device j. The latent space is edited by hiding the $z^{(t)}_{m\neq j}$ (we draw them randomly). The $z^{(t)}_j$ component remains unchanged. This forces the network to deduce $y^{(t)}_j$, the only useful value being the $z^{(t)}_j$ component. In practice, this masking operation is applied to $\frac{1}{8}$ of batch sequences.

In our specific use case, we adopt this method because knowing which $Z^{(t)}$ encodes the machine is crucial. However, in situations where such knowledge is unnecessary, we can bypass it by identifying the label encoded during test time, based on annotated data. However, the results obtained using this method or with TC (Total Correlation) don't contribute much to disentangling; the only significant improvement is in alignment.

The overarching training objective for the sequence-to-sequence model, incorporating residual KL in each layer $l = L, L-1, \dots 1$ as discussed in our proposed method above (Section 3.2), can be summarized as follows:

$$\mathcal{L}(\gamma, \beta, \delta; \theta, \phi, \psi) = \mathcal{L}_{rec} + \beta\, \mathcal{L}_{\mathrm{KL}} + \gamma\, \mathcal{L}_{TC} \qquad (9)$$

Here, we have a hyperparameter β_{KL} to balance the reconstruction loss and KL losses and γ to balance the disentangling effect of TC.

4. Numerical Experiments

4.1. Datasets and Baselines

We conducted experiments on two publicly available datasets, namely UK-DALE (Kelly & Knottenbelt, 2015) and REDD (Kolter & Johnson, 2011). The dataset UK-DALE (Kelly & Knottenbelt, 2015) consists of 5 dwellings with a varying number of sub-metered devices and includes aggregate and individual aggregate and individual equipment-level power measurements, sampled equipment, sampled at 1/6 Hz. We have focused our analysis on three analysis on three specific pieces of equipment: A Fridge, Washing Machine, and Oven. Similarly, for REDD we recover all 6 dwellings. **In order to assess the generalizability of the generalization of the models, we trained the models on the dataset REDD and we tested them on the dataset UK-DALE and then reversed this procedure.**

4.2. Architecture

Our model uses a bi-directional encoder, which processes the input data in a hierarchical manner to produce a low-resolution latent code that is refined latent code that is refined by a series of oversampling layers. This code is then refined by a series of oversampling layers in *Residual Decoders* blocks, which progressively increases the resolution. The residuals consist of a set of (*Batch Instance Normalization BN, conv(1x1), BN+Relu, conv (3x3), BN+Relu, and finally conv(1x1)*). The use of *Residual layers* allows us to efficiently capture semantic features in time series, while the temporal attention ensures the temporal correlation over latent space. In our architecture, the smallest dimension of Z is set to $\mathbb{R}^{(M+1)\times L}$ with $d_z = 16$ and $M = 3$, it is the number of devices to be separated in a mixed sequence of size $\tau_0 = 256$ *(A detailed explanation and results for the $M = 7$ case are provided in the supplementary material).*. To perform sampling, we conduct two tests: one for the classical case with a Gaussian prior using Monte Carlo with $k = 1$, as described in (Chen et al., 2018c), and another for

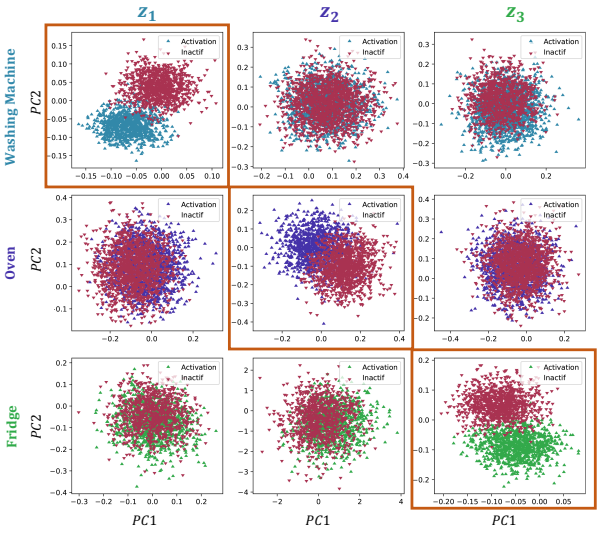

Figure 3. Each row shows the latent representation of at least one activated appliance (**Washing Machine**, **Oven**, and **Fridge** from top to bottom). The columns correspond to the $Z^{(t)}_m$ component of the structured latent variable $Z^{(t)}$ corresponding to the activation of m devices.

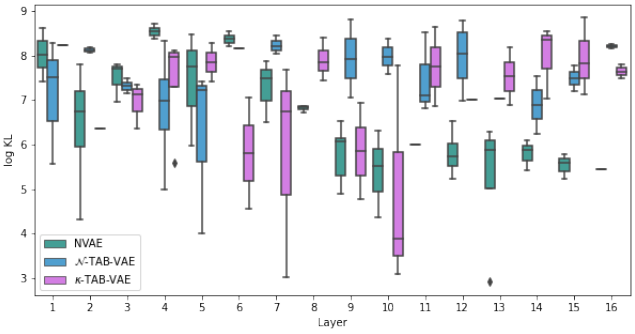

Figure 4. Comparison of KL divergence in each layer for $\mathcal{N}$-TAB-VAE, κ-TAB-VAE (refer to the case of using vMF distribution) and NVAE with $L = 16$.

the case with a von Mises-Fisher (vMF) distribution with $\kappa = 1$ using acceptance-rejection sampling. The implementation for the vMF distribution is inspired from (Davidson et al., 2018).

4.3. Performance and Informativity of TAB

Disentanglement Control through γ The findings are presented in Table (2). The Mean Squared Error (MSE) criterion represents the data attachment metric, averaged over the test dataset. To detect the presence of specific appliances, we employ thresholding identical to the method used in (Valenti et al., 2018). The detection performance is evaluated using the **F1** score on the test set. Remarkably, our approach surpasses the performance of S2S (Chen et al., 2018a), DAE (Valenti et al., 2018), S2P (Yang et al., 2021), and NVAE (Vahdat & Kautz, 2020) in terms of both **MSE** and **F1**. The relationship between latent variables z_j and their corresponding appliance states y_j is visually depicted in Figure (3). The latent vectors z_j were projected onto the two most significant dimensions of their Principal Component Analysis (PCA) representation.

Color-coding was applied to the points based on the machine's activation status. Each row corresponds to the activation or non-activation of a specific machine i (identified by color), while each column represents the visualization of z_j. Notably, points of different colors (active/inactive) on the diagonal are accurately distinguished by a line.

Impact of β **under Distribution Choice** When comparing the case where $\beta = 1$, meaning only Z is involved in the TAB equation, resulting in the context attention being ignored, we observe a slight improvement in the separation of the latent space. However, significantly better results are achieved for $\beta > 1$, as shown in Figure 3. We assert that our models perform exceptionally well when using the von Mises-Fisher (vMF) distribution compared to the normal distribution, as discussed in Section 3 (see Table 3). We studyin as well $\mathcal{L}_{KL}$ loss at each level. In Figure 4 TAB-VAE shows greater stability, indicating that all layers discover relevant information, which confirms our hypothesis in Section 3.2.

We claim that, at each level, the bottleneck is more flexible, and although we attempted a dynamic κ, it did not considerably improve the results. Therefore, setting κ as a constant value is one of the efficient solutions.

5. Conclusion and Perspectives

In this paper, we present a novel and interpretable approach for disaggregating load curves using an encoder-decoder architecture inspired by Variational Autoencoders (VAEs). Our novel approach is centered around enhancing the structure of the latent space through the utilization of a TAB cell, resulting in remarkable performance gains compared to current state-of-the-art methods. Additionally, our method facilitates the visualization of device activation states by effectively organizing the latent space during the learning process. As we move forward, our future work will focus on exploring the latent space in conjunction with more sophisticated features and examining the predictability of failure cases based on the latent representation.

6. Acknowledgements

The authors are grateful to the Area chair and three anonymous Reviewers for their valuable comments and interesting suggestions. This work was funded by One Tech, TotalEnergies Individual Fellowship. We would like to thank the Applied Data Scientist Team of One Tech for insightful discussions.

Table 1. Results on **UK-DALE** and **REDD** data: F1 score calculated on the test data, Mean Square Error (MSE) in $Watt^2$ calculated on the test data..

Machine	Dataset Test	GRU+	LSTM+	CNN	DAE	S2P	S2S	Bert4NILM	NVAE	$\mathcal{N}$-TAB VAE(Ours)	κ-TAB VAE (Ours)
							F1 (↑)				
Fridge	**UK-DALE**	81.52	81.62	81.59	81.80	83.73	83.73	83.73	90.10	**91.81**	90.25
	REDD	82.34	82.39	82.37	81.90	86.96	87.09	86.96	93.23	94.25	**94.81**
Washing Machine	**UK-DALE**	82.03	82.10	82.08	83.99	86.12	86.12	86.12	87.32	**93.26**	92.72
	REDD	82.07	82.11	82.09	82.99	85.57	86.16	85.57	91.54	**93.07**	92.94
Oven	**UK-DALE**	82.34	82.43	82.40	86.08	83.63	83.63	83.63	81.13	93.77	**92.23**
	REDD	81.95	81.99	81.97	81.94	84.14	83.78	84.14	91.30	94.04	**94.57**
							MSE (↓)				
Fridge	UK-DALE	25.70	25.68	25.69	25.74	27.36	26.70	27.36	28.36	**19.55**	21.42
	REDD	25.49	25.47	25.48	26.56	30.68	26.56	30.68	21.18	**19.48**	20.92
Washing Machine	UK-DALE	25.78	25.76	25.77	25.63	28.92	24.72	28.92	21.12	**18.33**	19.84
	REDD	25.59	25.57	25.58	25.34	28.40	24.78	28.40	23.22	**18.31**	19.65
Oven	UK-DALE	25.61	25.59	25.60	25.46	25.28	23.98	25.28	22.18	**19.30**	20.65
	REDD	25.45	25.43	25.44	25.42	25.04	23.94	25.04	20.78	19.82	**19.55**

References

Chen, K., Wang, Q., He, Z., Chen, K., Hu, J., and He, J. Convolutional sequence to sequence non-intrusive load monitoring. *the Journal of Engineering*, 2018(17):1860–1864, 2018a.

Chen, R. T., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018b.

Chen, R. T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018c.

Ciancetta, F., Bucci, G., Fiorucci, E., Mari, S., and Fioravanti, A. A new convolutional neural network-based system for nilm applications. *IEEE Transactions on Instrumentation and Measurement*, 2021. doi: 10.1109/TIM.2020.3035193.

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

Kelly, J. and Knottenbelt, W. The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes. *Scientific data*, 2, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kolter, J. Z. and Johnson, M. J. Redd: A public data set for energy disaggregation research. In *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, volume 25, 2011.

Koublal, Ladjal S, B. D. l.-b. E. and Roueff, F. Xgen: A comprehensive archive and an explainable time series generation framework for energy. 2023. URL https://xgentimeseries.github.io.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, 2020.

Valenti, M., Bonfigli, R., Principi, E., and Squartini, S. Exploiting the reactive power in deep neural models for non-intrusive load monitoring. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018.

Yang, M., Li, X., and Liu, Y. Sequence to point learning based on an attention neural network for nonintrusive load decomposition. *Electronics*, 2021.

A. Extended Discussion and Supporting Proofs

A.1. Proof of the Reconstruction Loss Proxy for Energy Disaggregation

Proof. The loss function of $\log p_\theta(X^{(t)}|Z^{(t)})$ can be written with Gaussian distribution as:

$$\log p_\theta(Y^{(t)}|Z^{(t)}) = \log \sigma_\theta(Z^{(t)}) + \frac{1}{2}\log 2\pi \tag{10}$$

$$+ \frac{1}{2}\frac{|Y^{(t)} - \mu_\theta(Z^{(t)})|^2}{\sigma_\theta(Z^{(t)})^2} \tag{11}$$

$$\propto ||Y^{(t)} - \mu_\theta(Z^{(t)})||^2 \tag{12}$$

where $\mu_\theta(Z^{(t)})$ and $\sigma_\theta(Z^{(t)})$ are neural networks that reconstruct $Y^{(t)}$ from latent representations. By the above equation, maximizing the reconstruction loss is regarded as minimizing the Euclidean distance between inputs and reconstructions. Thus, we opt to optimize the reconstruction loss by finding a way that imposes closer distances between raw input X and reconstructed $\hat{X}^{(t)} = \sum_{m=1}^{M} \hat{y}_m^{(t)} + \hat{\xi}^{(t)}$ from outputs $Y = \{\hat{y}_m\}_{m=1}^{M}$. We assume that the noise $\hat{\xi}^{(t)}$ is an interference term. □

A.2. Proof of Disentangled Representation using TC

To prove the equivalence between minimizing the KL divergence $KL(p_\phi(Z)||\prod_j p_\phi(z_j))$ and maximizing the Total Correlation (TC) loss $\mathcal{L}_{TC} = \mathbb{E}[\log(D_\psi(z_{\text{permuted}}))] + \mathbb{E}[\log(1 - D_\psi(Z))]$, we can follow the steps outlined below.

Start with the KL divergence expression:

$$KL(p_\phi(Z)||\prod_j p_\phi(z_j)) = \mathbb{E}_{p_\phi(Z)}\left[\log\left(\frac{p_\phi(Z)}{\prod_j p_\phi(z_j)}\right)\right]$$

Introduce the discriminator function $D_\psi(Z)$ to estimate the probability of a given Z being real (from the true posterior $\prod_j p_\phi(z_j)$). Rewrite the inequality using Jensen's inequality. Now, introduce the concept of a permuted latent variable z_{permuted}, which is a shuffled version of the original Z.

Rewrite the inequality in terms of the discriminator function $D_\psi(z_{\text{permuted}})$:

$$KL(p_\phi(Z)||\prod_j p_\phi(z_j)) \geq \mathbb{E}_{p_\phi(Z)}\left[\log \frac{D_\psi(z_{\text{permuted}})}{\mathbb{E}_{\prod_j p_\phi(z_j)}[1 - D_\psi(Z)]}\right]$$

Observe that $\log(D_\psi(z_{\text{permuted}}))$ can be interpreted as the log-likelihood of z_{permuted} being real (from the true poste-

rior), and $\log(1 - D_\psi(Z))$ can be interpreted as the log-likelihood of Z being fake (from $p_\phi(Z)$).

By maximizing the above expression, we aim to train the discriminator to accurately distinguish between real (permuted) and fake (original) samples. This corresponds to the Total Correlation (TC) loss, which encourages statistical independence between the components of Z.

Therefore, we have shown that minimizing the KL divergence is equivalent to maximizing the Total Correlation (TC) loss expressed in terms of the discriminator function $D_\psi(z_{\text{permuted}})$ and $D_\psi(Z)$.

A.3. Inference procedure

Algorithm 1 Bottom-Up Pass Path

Require: Set $h_{L+1} \equiv \mathbf{X}^{(t)}$ and $\mathbf{K}_q \equiv 0$ for $l = L, L-1, \ldots, 1$.
for $l = L, L-1, \ldots, 1$
do
$[h^{(l)}, \mathbf{K}_q] \leftarrow \mathbf{T}_{ql}(h_{(l+1)} \oplus \mathbf{K}_q)$.
end for
Return:
$h \triangleq \{h^{(l)}\}_L^{l=1}$: Extracted hidden features from data.
$\mathbf{K}_q \triangleq \{\mathbf{K}_q^l)\}_L^{l=1}$: Extracted Keys.

A.4. From Gaussian residual to vMF residual parametrization

vMF with TAB. In our proposed method, we implement also a von Mises-Fisher (vMF) distribution as both the prior and variational posterior within our TAB cells to compare it with Gaussian distribution results. The prior, denoted as vMF$(\cdot, \kappa = 0)$, is represented by a uniform distribution. Since the true posterior $p_\theta(z|x)$ is intractable, we approximate it with a variational posterior $\mathbf{Q}_\phi(z|x) =$ vMF$(z; \mu, \kappa)$, where the mean direction μ is obtained from encoding neural networks, and κ is considered a fixed constant, as depicted in Figure 2-b. Before implementing a VAE, we derive the KL divergence expression to optimize ELBO and provide a sampling algorithm using the reparameterization trick (Kingma & Ba, 2014).

Lemma A.1. ***KL divergence, case where*** $\kappa = 0$**.**

With vMF$(\cdot, 0)$ as our prior, the KL divergence is:

$$\begin{aligned} KL(vMF(\mu,\kappa)||vMF(\cdot,0)) &= \kappa \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)} \\ &+ \left(\frac{d}{2} - 1\right)\log\kappa - \frac{d}{2}\log(2\pi) \\ &- \log I_{d/2-1}(\kappa) + \frac{d}{2}\log\pi \\ &+ \log 2 - \log\Gamma\left(\frac{d}{2}\right) \end{aligned}$$

We utilize Lemma.A.1 to compute both KL terms in Eq.**??**. As observed, this KL term depends solely on κ and not on μ. Since we treat κ as a fixed hyperparameter, this term remains constant in our model, thereby preventing KL collapse. For the $\mathcal{N}$-TAB, the KL divergence in the objective function tends to pull the posterior towards the origin-centered prior, leading to challenging optimization. However, for the vMF VAE, given fixed κ, such vacuous states do not exist, and μ can freely vary.

Sampling from the von Mises-Fisher (vMF) Distribution For sampling from the vMF distribution, we adopt the sophisticated rejection sampling scheme based on (Davidson et al., 2018). This intricate process involves sampling a "change magnitude" denoted by w, which, together with a randomly selected unit vector v tangent to the hypersphere at μ, yields the final sample Z using the expression $z = w\mu + v\sqrt{1 - w^2}$.

A fascinating aspect of this sampling approach is its independence from the vMF mean parameter μ for both the randomly sampled unit vector v and the "change magnitude" w. This unique property allows for efficient computation of gradients of Z with respect to μ when needed.

A.5. Ordering and Alignment by Masking

Our inspiration is derived from information theory, specifically proposition.A.2, which provides valuable insight. When the information of $\mathbf{X}^{(t)}$ is concealed, it exists within the latent space $Z^{(t)}$ as well as in $\mathbf{Y}^{(t)}$. Leveraging this understanding, we have devised a method to instruct our model to utilize this property for the purpose of masking and aligning the latent space with a specific device.

Proposition A.2. *Assuming $\mathbf{X}^{(t)}$, Z, and $\mathbf{Y}^{(t)}$ form a Markov chain, $\mathbf{X}^{(t)} \rightarrow Z^{(t)} \rightarrow \mathbf{Y}^{(t)}$, where $p(\mathbf{Y}^{(t)}|\mathbf{X}^{(t)}, Z^{(t)}) = p(\mathbf{Y}^{(t)}|Z^{(t)})$, the data processing inequality ensures that $I(\mathbf{X}^{(t)}; z) \geq I(\mathbf{X}^{(t)}; \mathbf{Y}^{(t)})$. If $Z^{(t)}$ is a deterministic or stochastic function of $\mathbf{X}^{(t)}$, it cannot contain more information about $\mathbf{Y}^{(t)}$ than $\mathbf{X}^{(t)}$ itself.*

Proof. One can apply the chain rule for mutual information to obtain two different decompositions of

Algorithm 2 Organizing and Alignment of Z by Masking

Require: Sequence $X(t)$, Masking probability p
Ensure: Latent space representation $Z^{(t)}$
  for t in $\mathcal{B}$ **do**
    Select device j randomly
    Sequence $y_j^{(t)}$ corresponding to device j from $X^{(t)}$
    Set $Z_m^{(t)}$ to random for all components except $Z_j^{(t)}$
    if Random() $< p$ **then**
      Apply masking to $Z^{(t)}$ by hiding the values $Z_m^{(t)}$
    end if
    Compute the output $\hat{y}_j^{(t)}$ corresponding to device j
  end for

$I(\mathbf{X}^{(t)}; \mathbf{Y}^{(t)}, Z^{(t)})$: $I(\mathbf{X}^{(t)}; Z^{(t)}) + I(\mathbf{X}^{(t)}; \mathbf{Y}^{(t)} \mid Z^{(t)}) = I(X; \mathbf{Y}^{(t)}, Z^{(t)}) = I(\mathbf{X}^{(t)}; \mathbf{Y}^{(t)}) + I(\mathbf{X}^{(t)}; Z^{(t)} \mid \mathbf{Y}^{(t)})$
By the relationship $X \to Y \to Z$, we know that X and Z are conditionally independent, given $\mathbf{Y}^{(t)}$, which means the conditional mutual information, $I(\mathbf{X}^{(t)}; Z^{(t)} \mid \mathbf{Y}^{(t)}) = 0$. The data processing inequality then follows from the non-negativity of $I(X; Y \mid Z) \geq 0$. □

A.6. Computation

Table 2 presents a comparison of the computational requirements for training different VAE models, including NVAE (Normal VAE), $\mathcal{N}$-TAB-VAE (Normal TAB-VAE), and κ-TAB-VAE (Kappa TAB-VAE) on the Uk-dale dataset. The training is conducted using the XGen framework.

The table shows the batch size per GPU, the number of GPUs utilized for training, and the corresponding training time in hours for each model. The batch size for all models is set to 128, and four GPUs are used in parallel for training in each case.

As observed from the table, the $\mathcal{N}$-TAB-VAE and κ-TAB-VAE models exhibit longer training times compared to NVAE. This indicates that the additional computational cost associated with computing attention scores in the $\mathcal{N}$-TAB-VAE and κ-TAB-VAE models is offset by the benefits of having a smaller number of stochastic layers in the hierarchical architecture without compromising the generative capacity of the models.

This information provides valuable insights into the computational efficiency and trade-offs among these state-of-the-art VAE models when applied to the Uk-dale dataset.

A.7. Impact of window parameter τ

To perform Non-Intrusive Load Monitoring (NILM) effectively, it is crucial to select an appropriate window time series. This involves determining a time interval for analyzing energy consumption data that allows for the detection and classification of individual appliance activities. The chosen window should strike a balance between being long enough to capture complete appliance activity cycles and short enough to avoid overlaps with other activities or periods of inactivity. The optimal window size depends on factors such as the energy meter's sampling rate, the number and types of appliances being monitored, and the specific NILM algorithm employed. Experimentation and optimization may be necessary to identify the ideal window size for a specific NILM application. In our study, we tried to detect the consumption of the washing machine, which averages 3 to 4 hours of use per cycle. Therefore, we chose a window of 4h30, equivalent to 256-time steps of 60 seconds. In addition, we've noticed that a window of 128 and 300 steps doesn't detect the washing machine.

Table 2. We compare the computational requirements for training TAB-VAE and NVAE models on the Uk-dale dataset. The training is performed using Nvidia A100 GPUs, each equipped with 80GB of memory. We utilize the XGen (Koublal & Roueff, 2023) framework for conducting the training process.

Model	Batch/GPU	# GPUs	Time (hour)
NVAE	128	4	68
$\mathcal{N}$-TAB-VAE	128	4	84
κ-TAB-VAE	128	4	152

A.8. Optimization

In all of our experiments, we used the Adam optimizer with an initial learning rate of 10^{-3} and a cosine decay of the learning rate. We also reduced the learning rate to 7×10^{-4} to increase the stability of the training and applied an early stop after 5 iterations. We set $\alpha = 0.5$ and $\beta = 2.5$ after a grid search on the best convergence of the model on the validation data.

A.9. TAB-VAE training results for the $M = 7$, $\tau = 256$ and $n = 7$ case on REDD, Uk-Dale and REFIT

Table 3. Performance on **UK-DALE**, **REDD** and **REFIT** datasets, "–" denotes the unknown result due to the high complexity of the corresponding method. F1 score (higher is better), MAE, and MSE (lower is better) are computed on the test set. For each model, the best configuration is the one achieving the lowest MSE on the validation set.

Dataset	Method	Metric	Fridge	Clothes dryer	Stove	Washing Machine	Dishwasher	Oven
UK-DALE	DAE	F1 (↑)	80.57	81.37	81.47	83.01	81.41	81.80
	S2S		83.99	86.08	83.79	84.85	83.28	83.61
	S2P		83.73	86.12	83.23	84.56	83.28	83.63
	NVAE		**91.71**	92.14	92.30	91.63	**92.32**	93.11
	TAB-VAE		91.81	**93.26**	**92.99**	**92.67**	92.21	**93.77**
	DAE	MAE (↓)	25.74	25.63	24.32	25.22	24.81	25.46
	S2S		26.70	24.72	30.05	25.56	24.49	23.98
	S2P		27.36	28.92	27.37	27.86	25.00	25.28
	NVAE		22.58	21.02	21.73	20.46	20.35	19.74
	TAB-VAE		**19.55**	**18.33**	**18.63**	**19.19**	**17.49**	**19.30**
	DAE	MSE (↓)	243.52	244.08	245.74	244.07	243.70	243.18
	S2S		-	-	-	-	-	-
	S2P		-	-	-	-	-	-
	NVAE		163.01	162.80	162.58	163.28	163.04	171.34
	TAB-VAE		**164.22**	**161.58**	**161.87**	**156.77**	**152.28**	**152.02**
REDD	DAE	F1 (↑)	82.99	81.94	82.01	82.51	81.61	81.90
	S2S		87.09	86.16	83.43	84.83	83.30	83.78
	S2P		86.96	85.57	83.52	85.08	83.97	84.14
	NVAE		93.23	92.29	91.53	91.54	**92.69**	92.30
	TAB-VAE		**94.25**	**93.07**	**93.33**	**92.90**	92.82	**94.04**
	DAE	MAE (↓)	26.56	25.34	24.70	24.99	25.30	25.42
	S2S		26.56	24.78	29.78	25.78	24.04	23.94
	S2P		30.68	28.40	27.65	27.43	24.24	25.04
	NVAE		22.74	21.35	21.74	19.85	20.56	19.99
	TAB-VAE		**19.48**	**18.33**	**19.16**	**18.75**	**17.25**	**19.55**
	DAE	MSE (↓)	243.53	244.73	245.34	244.62	243.74	243.91
	S2S		-	-	-	-	-	-
	S2P		-	-	-	-	-	-
	NVAE		**163.61**	162.88	162.81	163.24	162.32	170.97
	TAB-VAE		163.63	**161.78**	**162.06**	**157.19**	**152.21**	**151.65**
REFIT	DAE	F1 (↑)	80.80	81.18	81.30	82.27	82.08	82.27
	S2S		83.76	85.45	83.77	84.49	83.75	84.35
	S2P		83.64	85.37	83.67	84.70	83.54	84.07
	NVAE		**91.46**	92.06	92.07	92.00	**93.13**	93.11
	TAB-VAE		92.17	**93.15**	**92.89**	**92.82**	92.64	**94.04**
	DAE	MAE (↓)	25.40	25.11	23.85	24.84	25.20	25.59
	S2S		27.34	24.84	29.50	26.05	24.26	24.20
	S2P		26.98	28.36	27.23	27.80	24.87	25.19
	NVAE		21.88	20.96	22.17	19.72	19.76	19.75
	TAB-VAE		**18.81**	**18.83**	**19.47**	**19.05**	**17.13**	**19.15**
	DAE	MSE (↓)	243.67	244.17	245.34	244.51	243.41	243.71
	S2S		-	-	-	-	-	-
	S2P		-	-	-	-	-	-
	NVAE		**163.57**	**163.08**	**162.12**	163.19	162.48	170.85
	TAB-VAE		163.85	161.74	162.57	**156.66**	**151.71**	**151.76**

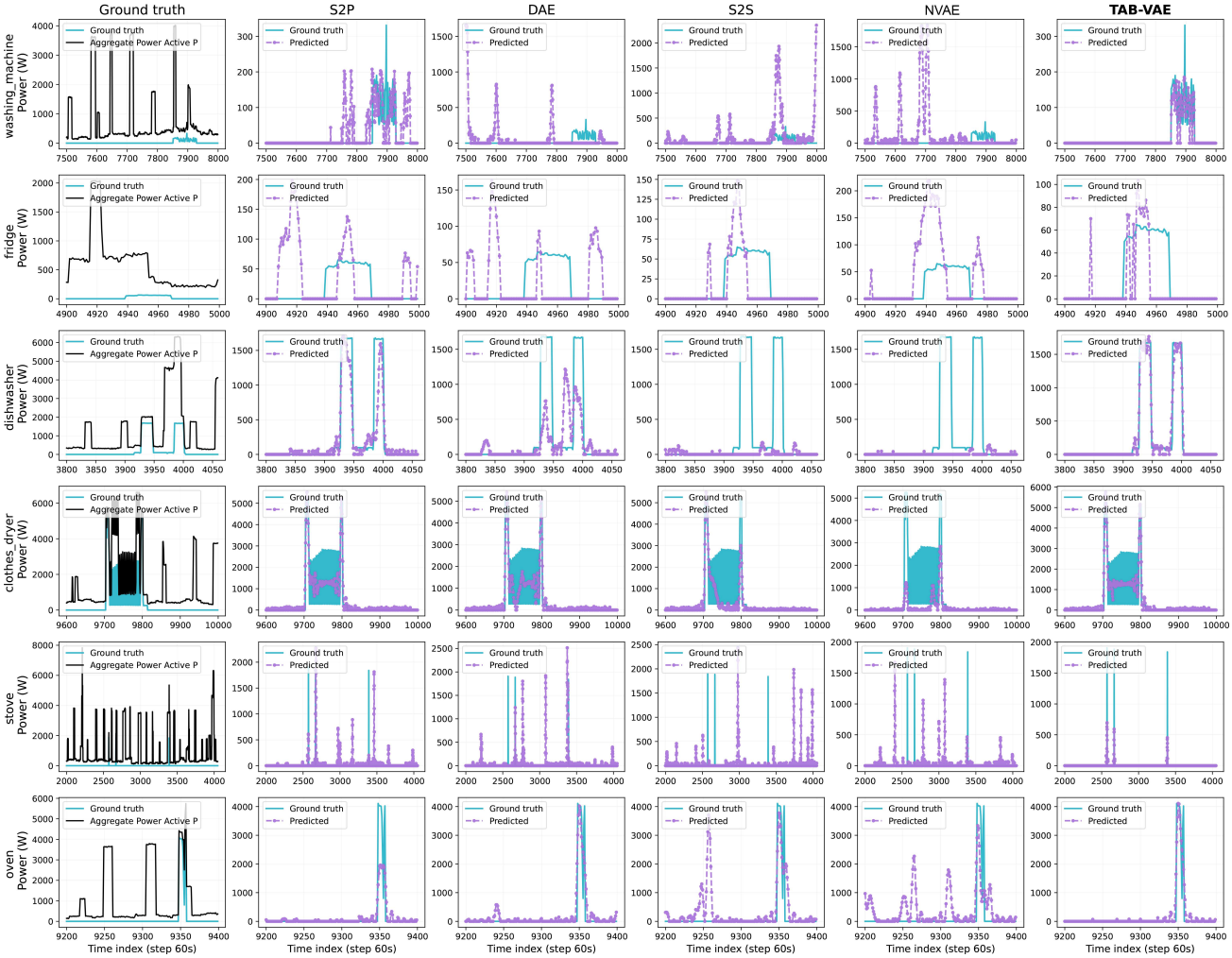

Figure 5. The results of the disaggregation models tested on **UK-DALE** dataset are presented from left to right. The models included in the comparison are **S2P** based on DeepAR, **DAE**, S2S, **NVAE**, and **TAB-VAE** (our model). Each row corresponds to a different appliance, from top to bottom: washing machine, fridge, dishwasher, clothes dryer, stove, and oven .