DISENTANGLING TIME SERIES REPRESENTATIONS VIA CONTRASTIVE BASED l-VARIATIONAL INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning disentangled representations is crucial for Time Series, offering benefits like feature derivation and improved interpretability, thereby enhancing task performance. We focus on disentangled representation learning for home appliance electricity usage, enabling users to understand and optimize their consumption for a reduced carbon footprint. Our approach frames the problem as disentangling each attribute's role in total consumption (e.g., dishwashers, fridges, etc). Unlike existing methods assuming attribute independence, we acknowledge real-world time series attribute correlations, like the operating of dishwashers and washing machines during the winter season. To tackle this, we employ weakly supervised contrastive disentanglement, facilitating representation generalization across diverse correlated scenarios and new households. Our method utilizes innovative l-variational inference layers with self-attention, effectively addressing temporal dependencies across bottom-up and top-down networks. We find that **DIoSC** (**D**isentanglement and **I**ndependence-of-**S**upport via **Co**ntrastive Learning) can enhance the task of reconstructing electricity consumption for individual appliances. We introduce TDS *(Time Disentangling Score)* to gauge disentanglement quality. TDS reliably reflects disentanglement performance, making it a valuable metric for evaluating time series representations. Code available at https://anonymous.4open.science/r/DIoSC.

1 INTRODUCTION

Disentangled representation learning is crucial in various fields like computer vision, speech processing, and natural language processing (Bengio et al., 2014). There have been efforts to learn disentangled time series representation (Woo et al., 2022; Yao et al., 2022), with the aim to improve generalization, robustness, and explainability. We call a representation disentangled when identified attributes in the data are specifically coded in the structure of its latent units. How this can be achieved remains an open research question. In (Locatello et al., 2019), it is shown that disentangling requires some kind of supervised learning and inductive bias. Moreover, standard methods such as β-VAE, TCVAE, and NVAE (Higgins et al., 2016) rely on the too stringent assumption of statistical independence among ground truth attributes. In real-world time series, attributes are often correlated. In the application of this study, the attributes correspond to the contributions of specific devices in an aggregated consumption signal. We illustrate the correlation of the attributes in Fig 1, where green boxes contain typical consumption for their respective appliances and purple ones show consumption profiles that are correlated to those of the other appliances. Existing methods in this context, such as (Yang et al., 2021a; Woo et al., 2022), often assume independent attributes. In this work, we explore a broader aspect of disentangling time series by relaxing the assumption of statistical independence in the latent representation.

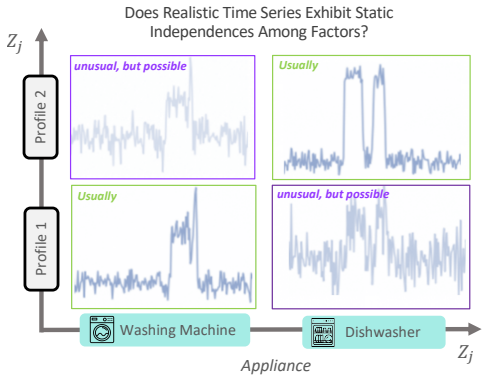

Figure 1: Real-world data often showcases attributes exhibiting strong positive correlation.

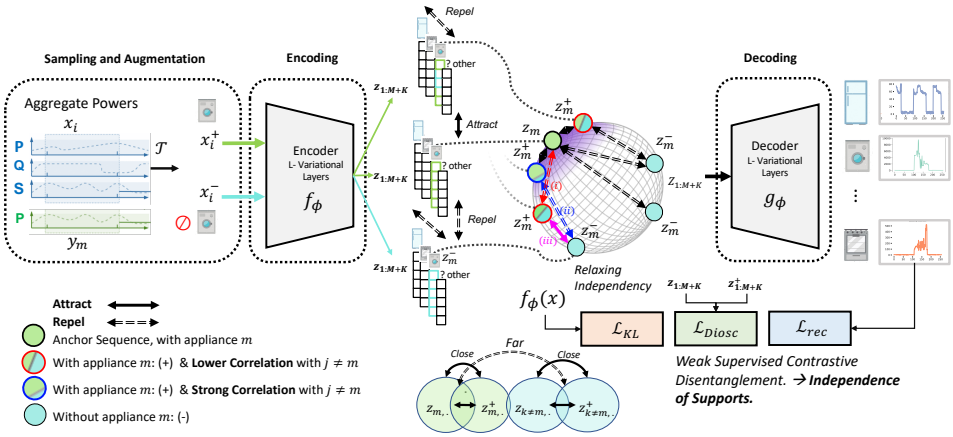

Figure 2: Our framework relies on *Disentanglement and Independence-of-Support via Contrastive (Diosc)* to encourage the latent space $\mathbf{z}$ to exhibit a support factorization, permitting some level of dependency. Representations of positive pairs attract each other, while negative repels their corresponding representations. The latent attributes exhibit causal correlations (Shanmugam, 2018), our framework allows for scenarios where unlikely (but exist in data) combinations occur ((i) and (ii) leading to the existence of (iii)). It's worth noting that forcing strict statistical independence does not prevent these cases.

Let us more precisely detail a key application for time series disentanglement : household energy consumption disaggregation, also known as Non-Intrusive Load Monitoring (NILM). Given only the main consumption of a household $\mathbf{x} \in \mathbb{R}^{c \times T}$, seen as a c-variate time series observed at times $t = 1, \dots, T$, the NILM algorithm identifies the active consumption $y_m \in \mathbb{R}^T$ of each operating appliance m. Such a task has received a growing interest and still raises unsolved problems. The fact that many households rely on past bills to adjust future energy use underscores the importance of energy disaggregation algorithms in reducing carbon footprints. Recent work (Bucci et al., 2021; Yang et al., 2020; Nalmpantis & Vrakas, 2020) hold promising results, yet challenges in generalization and robustness persist. In this work, we tackle NILM with a disentanglement perspective and study the disentanglement as well as the downstream disaggregation task performances.

Contributions and Main results. Our approach stands out by relying on weak contrastive learning using support factorization on the prior (rather than strict statistical independence) and Attentive *l*-Variational Inference. We evaluate our method qualitatively and quantitatively across various datasets with ground-truth labels, examining the generalization capabilities of the learned representations on correlated data. In summary:

1. We define a new regularization term, DIoSC, whose goal is to address latent space misalignment issues and to preserve disentanglement. This is achieved by promoting specific *Pairwise (dis-)similarities* over the latent sub-variables (c.f. §4).
2. Our experiments across three datasets and diverse correlation scenarios demonstrate that Diosc significantly enhances robustness to attribute correlation, yielding up to a **+61.4%** for reconstruction and **+21.7%** improvement in disentanglement metrics (RMIG, DCI, and TDS) compared to state-of-the-art methods (c.f. §5.3).
3. We propose *l*-variational-based self-attention for extracting high semantic representations from time series, ensuring complex representations without temporal locality.
4. To evaluate disentangelement we proposed Time Desintengling score, alig with the performance in downstream task. We implemented our framework in a user-friendly library, making it the first time-series disentanglement framework.[1].

2 Related Work

On The Non-Intrusive Load Monitoring and Representation Learning. Recent work (Bucci et al., 2021; Nalmpantis & Vrakas, 2020) has produced promising results for separation source power. Nevertheless, they encounter challenges related to generalization and robustness when confronted with out-of-distribution scenarios. Several approaches have been suggested to address these challenges. Some methods tackle them through either transfer learning or by enhancing the learned representations for each individual appliance. Exploring ways to enhance representation learning in this field has been the focus of recent studies (Woo et al., 2022; Vahdat & Kautz, 2021; Maaløe

[1] https://anonymous.4open.science/r/DIoSC

et al., 2019). However, achieving an informative and disentangled representation remains an open and challenging question. Existing models, like RNN-VAE (Chung et al., 2015) for sequential data and D3VAE (Li et al., 2023), assume statistically independent attributes. This assumption hampers their performance on real-world data and makes them less applicable to out-of-distribution scenarios. Developing models that effectively capture informative and disentangled representations in a realistic and versatile manner continues to be a significant challenge.

Time Series Disentanglement in the Realm of Correlated Attributes. Traditional methods for time series disentanglement often emphasize enforcing statistical independence among representation dimensions (Do & Tran, 2021; Klindt et al., 2020), even when dealing with highly correlated data. In recent computer vision disentanglement methods, there has been an exploration of using auxiliary information to improve identifiability, moving away from the assumption of statistical independence (Roth et al., 2023). However, both (Träuble et al., 2021; Roth et al., 2023) point out the limitations of this approach due to non-identifiability. Another study by (Wang & Jordan, 2022a) proposes support factorization for disentanglement from a causal perspective, incorporating a Hausdorff objective akin to (Roth et al., 2023). In our unique approach, we tackle time series disentanglement without explicit auxiliary variables or prior models. Instead, we achieve pairwise factorized support through contrastive learning, departing from the traditional independence assumption. This method pioneers disentanglement in correlated time series by emphasizing independence-of-support through contrastive learning during training. This contrasts with methods like (Ren et al., 2021), where representation discovery relies on contrastive learning of pre-trained generative models with assumed independence factorization during training. To our knowelege, we are the first to use weakly supervised contrastive to disentangle time series with an *l*-VAE in correlated scenarios.

3 Problem Statement

We consider a c-variate time series observed at times $t = 1, \dots, T$, we denote by $\mathbf{x} \in \mathbb{R}^{c \times T}$ the $c \times T$ resulting matrix with rows denoted by $x_1, \dots, x_c$. Each row can be seen as a univariate time series. In the context of electric load applications, we have $c = 3$ and x_1, x_2, x_3 respectively represents the sampled active power, the sampled reactive power, and the sampled apparent power. Our goal is to recover the following decomposition of the active power x_1:

$$x_1 = \sum_{m=1}^{M} y_m + \xi \ , \tag{1}$$

where $y_m \in \mathbb{R}^T$ denotes the contribution of the m-th electric device, among the total of M devices identified within the household, $\xi \in \mathbb{R}^T$ contains the contribution of unknown appliances and additive noise. We denote by $\mathbf{y}$ the $M \times T$ matrix obtained by stacking active power devices' contributions. The Non-Intrusive Load Monitoring (NILM) mapping, denoted as $\mathbf{x} \mapsto \mathbf{y}$, is typically learned from a training set $\mathcal{X} = \{\tilde{\mathbf{x}}_i\}_{i=1}^N$, where each $\tilde{\mathbf{x}}_i = (\mathbf{x}_i, \mathbf{y}_i)$ represents a pair of input-output samples used for training purposes.

Our approach belongs to the general framework of Variational Auto-Encoders VAEs, and thus relies on two main ingredients: 1) a generative model (p_θ) based on a latent variable, and a decoder g_θ; 2) a variational family (q_ϕ), which approximates the conditional density of the latent variable given the observed variable based on an encoder f_ϕ. In a VAE, both (unknown) parameters θ and ϕ are learnt from the training set $\mathcal{X}$. A key idea for defining the goodness of fit part of the learning criterion is to rely the **E**vidence **L**ower **B**ound (ELBO), which provides a lower bound on (and a proxy of) the log-likelihood

$$\log p_\theta(\tilde{\mathbf{x}}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\tilde{\mathbf{x}})} \left[\log p_\theta(\tilde{\mathbf{x}}|\mathbf{z})\right] - \mathrm{KL}(q_\phi(\mathbf{z}|\tilde{\mathbf{x}}) \parallel p(\mathbf{z})) \ , \tag{2}$$

where we denoted the latent variable by $\mathbf{z}$, defined as a $d_z \times (M + K)$ matrix and p denotes its distribution. The use of ELBO goes back to traditional variational Bayes inference. Furthermore, in the context of VAE, the inference network is defined as q_ϕ and the generative model as p_θ through an encoder/decoder pair of neural networks (f_ϕ, g_θ). In the NILM, f_ϕ provides an approximation of $\mathbf{z} = \{z_1, \dots, z_{M+K}\}$ from $\mathbf{x}$ while g_θ provides an approximation of $\mathbf{y}$ from $\mathbf{z}$. A standard choice in a VAE is to rely on Gaussian distributions and, for instance, to set $q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu(\mathbf{x}, \phi), \sigma^2(\mathbf{x}, \phi))$, where $\mu(\mathbf{x}, \phi)$ and $\sigma^2(\mathbf{x}, \phi)$ are the outputs of the encoder f_ϕ. As Discussed in Section 1, various criterion functions such as β/TC/Factor/DIP-VAE have

been introduced, aiming to learn a disentangled latent variable $\mathbf{z}$ and align it with the corresponding attributes. However, these methods typically assume statistical independence among attributes, leading to the assumption: $p(\mathbf{z}) = p(z_1) \dots p(z_{M+K})$. Contrary to this assumption, appliances are not used independently; rather, they used simultaneously, and their profiles may exhibit correlation (though less likely), thereby challenging the validity of Independent Factorization.

Definition 3.1 ***Independence-of-Support Factorization (IoS).*** *For a latent variable* $\mathbf{z} = \{z_1, \dots, z_{M+K}\}$ *sampled from* $p(\mathbf{z})$*, if* $\mathcal{Z} = \mathcal{Z}_1 \times \dots \times \mathcal{Z}_{M+K}$*, where* $\mathcal{Z}$ *is the support of* $p(\mathbf{z})$*, and* $\mathcal{Z}_j$ *denotes the supports of marginal distributions of* z_j*, then* $z_1, \dots, z_{M+K}$ *are considered to have independent support.*

Although, work of (Wang & Jordan, 2022b; Roth et al., 2023), correlated attributes have been taken into account by replacing the independence constraint by Hausdorff Factorized Support (HFS) assumption to fulfill IoS (cf. def. 3.1) and its variants. However, the numerical stability challenge in incorporating a pairwise Hausdorff Distance (HDF) constraint prompts exploration of an alternative. We propose using a weakly supervised contrastive *Pairwise Similarity* penalty to ensure IoS constraint satisfaction and improved stability. Further details are provided in the following section.

4 Proposed Methods

To solve the NILM problem effectively in real-world scenarios, we seek an ideal disentangled representation. Rather than using individual decoders $g_\theta^{(m)}$ for each appliance $m = 1, \dots M$, we propose a single deep network for simultaneous and more efficient disentangling. To achieve this, we employ the following strategies: 1) **D**isentanglement and **I**ndependence-of-**S**upport via **Co**ntrastive Learning (**DIoSC**), which first promotes similarity between the latent representations z_m when the device m is present in both $\mathbf{x}_i$ and its augmentation while inducing dissimilarity in negative cases. Additionally, we relax independent factoring to IoS, recognizing the non-independence of device activations, rather than insisting on independent latent activations; 2) we propose using an Attentive *l*-Variational Auto-Encoder integrate self-attention mechanisms, proven effective in natural language processing (Vaswani et al., 2017), to enhance the model's ability to capture intricate patterns and achieve robust reconstruction. In the upcoming sections, we elaborate on each strategy and articulate the comprehensive objective function.

4.1 Disentangling based Support Factorization via Weakly Supervised Contrastive

In the realm of contrastive learning (Khosla et al., 2021), the fundamental concept involves compelling a model to generate distinct latent features for semantically different inputs. Extending this notion, our approach aims to imbue the latent variables z_m with the capacity to both reconstruct y_m and withstand the introduction of noise on appliance m. We also assign to z_m the task of being able to differentiate the status (active/non-active) of the machine m. Consider $\mathbf{x}$ a mixed power sample where we chose (at random) a machine m activated, and $\mathbf{z} = \{z_1 \dots z_m \dots z_{M+K}\}$ the corresponding latent mapped by f_ϕ. We generate a positive augmentation set for $\mathbf{x}$ using time-based augmentation (see §5.1), we denote the latent mapped for positive augmentation $\mathbf{z}^+ = \{z_1^+ \dots z_m^+ \dots z_{M+K}^+\}$. This resulting in positive pairs denoted as (z_m, z_m^+), where the corresponding latents z_m and z_m^+ are expected to be close to each other. Conversely, we characterize negative pairs (z_m, z_m^-) as pairs derived from a set of samples in which machine m is active in the first sample and inactive in the second (i.e., a sample where m is inactive). The standard contrastive loss in this setting would be:

$$\mathcal{L}_{\text{self}}^i = -\sum_m \log \frac{\exp(sim(z_m, z_m^+)/\tau)}{\sum_{\text{negative pairs}} \exp(sim(z_m, z_m^-)/\tau)}, \tag{3}$$

where $sim(\cdot, \cdot)$ denotes the cosine similarity, and $\tau \in \mathbb{R}^+$ is a temporal parameter to adjust scale. In this equation the sum in the numerator extends to all negative pairs generated for a sample i and its size depends on the batch size. In multiclass settings like NILM, the contrastive loss (Eq. 3) struggles when multiple samples share the same appliance latent z_m due to labeled data. A suggested generalization (Khosla et al., 2021) tackles labeled cases. However, drawbacks include the lack of $\mathbf{z}$ invariance and challenges with limited or noisy labels, especially in obtaining both negative and positive labels for time series.

Disentanglement, Invariant, and Axis-Alignment Latents. When the mth component of $\mathbf{z}$, denoted by z_m, remains invariant regardless changes in $\mathbf{x}$, z_m is meaningless and contains no information from $\mathbf{x}$. We consider that the latent space *aligned* when the variations of latent variables z_m only

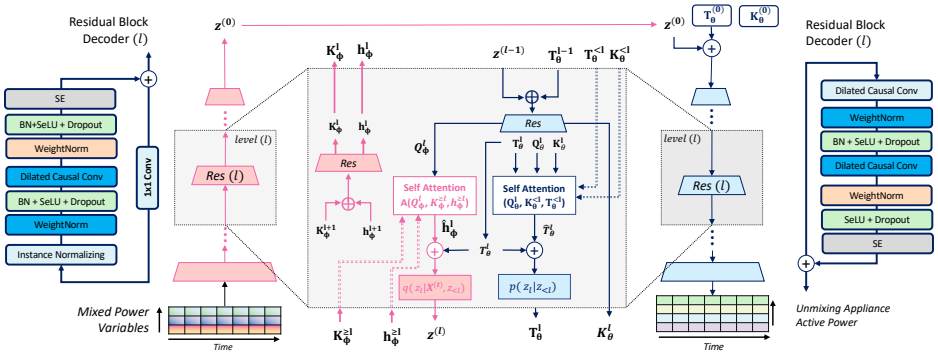

Figure 3: Performing Attentive l-Variational Inference entails processing power input $\mathbf{x} \in \mathbb{R}^{c \times T}$ through l levels of residual blocks ($\mathbf{Res}^{(l)}$), generating key and query feature maps. Parameters $[\mathbf{K}_q^{(L+1)},\ T_p^{(0)},\ \mathbf{K}_p^{(0)}]$ are initially set to zero, and $h^{(L+1)} \triangleq \mathbf{x}$.

have an influence on the m-th output of the decoder g_θ applied to $\mathbf{z}$. Both terms form the basis of the disentanglement principle. Thus, $\mathbf{z}$ is considered disentangled when there is a one-to-one correspondence between each ground truth y_m and the corresponding z_m in the representation. To uphold this despite the constraint of limited labels, hard to get both negative and positive in NILM, and without relying on static attribute independence, we leverage weak contrastive learning (Zimmermann et al., 2022; Zbontar et al., 2021), and adjusting it for disentanglement. The disentanglement loss we propose integrates two core components: (i) a component seeks to minimize information overlap between z_m and its negatives z_m^-; (ii) a term compelling similarity between z_m and its augmented z_m^+ accommodating potential changes to cover the variability factor of variation in ground-truth attributes, ensuring both *invariance*, *alignment* and enabling the discrimination task. Nevertheless, this is not sufficient to obtain an IoS. We next, demonstrate how this constraint could encompass an extra assumption relaxing the latent to IoS, as outlined below.

Assumption 4.1 ***Relaxing Independent Factorization to IoS.*** *Based on def. 3.1, we consider an empirical support $\mathcal{Z} \approx \mathbf{Z}$ where $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^b$, and b the mini-batch size. Ensuring IoS this implies that $\mathbf{Z}$ aligns with its Cartesian product $\mathbf{Z}^\times$. To achieve pairwise factorized support, we minimize sliced/pairwise contrastive, approximating $\mathbf{Z}$ and $\mathbf{Z}^\times = z_{:,1} \times \ldots \times z_{:,M+K}$, where $z_{:,m} \in \mathbb{R}^{b \times d_z}$.*

Building upon Assumption 4.1, which incorporates the *invariant* and *alignment* properties, we impose a constraint on a given mini-batch $\mathbf{Z}$ of size b. Specifically, we ensure that the elements $\mathbf{Z}_{:,m}$ are close to their corresponding augmented $\mathbf{Z}_{:,m}^+$ and far from any negative $\mathbf{Z}_{:,m}^-$, while simultaneously preserving the independence of the Support (IoS). This involves minimizing the distance between sets $\mathbf{Z}_{:(m,\neq m)}$ and $\mathbf{Z}_{:,m} \times \mathbf{Z}_{:,\neq m}$ for all appliances m. Owing the discriminating nature of contrastive learning over data, this IoS constraint can be met by focusing on contrastive learning $\mathbf{Z}_{:,m}$ and its augmentation $\mathbf{Z}_{:,m}^+$ without involving the Cartesian product $\times$ between support latent. Essentially, our approach contrastive effectively addresses the same instance discrimination task as when considering the Cartesian product over all possible combinations. This observation aligns with insights from a disentanglement causality perspective [2] (Wang & Jordan, 2022a). Further explanations are given in §5.3.

$$\mathcal{L}_{\text{Diosc}} = \underbrace{\eta \sum_{m} \sum_{\text{negative pairs}} \mathcal{D}(z_m, z_m^-)^2}_{(i)} + \underbrace{\sum_{m} \sum_{\text{positve pairs}} \left(1 - \mathcal{D}(z_m, z_m^+)\right)^2}_{(ii)}, \tag{4}$$

where $\mathcal{D}(\cdot,\cdot)$ is the cosine similarity distance. It is shown that both terms contribute equally to the improvement, i.e. $\eta = 1$. We denote $\mathcal{L}_{\text{Diosc-2}}$, case where the cartesian product still holds between latent variables for both positive and negative latents.

4.2 Attentive l-Variational Auto-Encoders

To avoid time locality during dimension reduction, and keep long-range capability we refer to an in-depth Temporal Attention with l-Variational layers. NVAE (Vahdat & Kautz, 2020; Apos-

[2] This study embraces a causal of representation learning, contrasting with DIoSC's relaxation of the independence assumption to Independence-of-Support (IoS) via contrastive.

tolopoulou et al., 2021) proposed an in-depth autoencoder for which the latent space $\mathbf{z}$ is level-structured and attended locally (Apostolopoulou et al., 2021), this shows an effective results for image reconstruction. We employ Temporal Multihead Self-attention (Vaswani et al., 2017) for constructing beliefs of variational layers, allowing effective handling of long context sequences. l-Variational Inference is illustrated in Fig. 3, where the construction of Temporal context $\hat{T}_p^{(l)}$ at level l relies on a preview contexts i.e $l-1$ denoted $T_p^{(<l)}$, query map $\mathbf{Q}_p^{(l)}$, and its key map $\mathbf{K}_p^{(<l)}$. This approach enables the model to attend to information from different representation subspaces at various scales. The use of Multihead self-attention aids in capturing diverse relationships and patterns. The detailed mechanism is given in Appendix. B. For the remainder, we assume that **DIoSC** uses attentive variational inference l. We adopt the Gaussian residual parametrization between the prior and the posterior. The prior is given by $p(\mathbf{z}_l|\mathbf{z}^{(<l)}) = \mathcal{N}(\mu(T_p^l,\theta), \sigma(T_p^l,\theta)$. The posterior is then given by $q(\mathbf{z}_l|\mathbf{x},\mathbf{z}^{(<l)}) = \mathcal{N}(\mu(T_p^l,\theta) + \mathbf{\Delta}\mu(\hat{T}_q^l,\phi), \sigma(T_p^l,\theta)\cdot\mathbf{\Delta}\sigma(\hat{T}_q^l,\phi))$ where $\mu(\cdot)$, $\sigma(\cdot)$, $\mathbf{\Delta}\mu(\cdot)$, and $\mathbf{\Delta}\sigma(\cdot)$ are transformations implemented as convolutions layers. Hence, the term $\mathcal{L}_{\mathbf{KL}}$ in Eq. 2 adding the residual and then the $\mathcal{L}_{\mathbf{KL}}$ is given by: $\mathcal{L}_{\mathbf{KL}}(\mathbf{x};\phi,\theta) = \mathcal{L}_{\mathbf{KL}}(\mathbf{x};\phi,\theta) + \sum_{l=1}^{(l)} 0.5 \times \left(\frac{\mathbf{\Delta}\mu^{(l)\mathbf{2}}}{\sigma^{(l)2}} + \mathbf{\Delta}\sigma^{(l)2} - \log\mathbf{\Delta}\sigma^{(l)2} - 1\right)$.

4.3 Setting Overall Objective Function

Our final objective function combines the regularization (Eq. 2) and the VAE loss (Eq. 2), which consisting of a reconstruction term $\mathcal{L}_{rec}$, a $\mathcal{L}_{\mathbf{KL}}$ term. We present balancing parameters, denoted as λ and β, with λ governing disentanglement and β balancing emphasis between the reconstruction and KL divergence terms.

$$\mathcal{L}(\mathcal{D},\phi,\theta) = \mathbb{E}_{\mathbf{X}\overset{b}{\sim}\mathcal{X}}\Big[\lambda\mathcal{L}_{\text{Diosc}} + \frac{1}{b}\sum_{\mathbf{x}\in\mathbf{X},\mathbf{y}\in\mathbf{Y}}\mathcal{L}_{rec}(\hat{\mathbf{y}},\mathbf{y};\phi,\theta) + \beta\mathcal{L}_{\mathbf{KL}}(\mathbf{x};\phi,\theta)\Big]. \tag{5}$$

4.4 How to evaluate disentanglement for Time Series?

Evaluating disentanglement in series representation is more challenging than established computer vision metrics. Existing time series methods rely on qualitative observations and predictive performance, while metrics like Mutual Information Gap (MIG) (Li et al., 2023) have limitations with continuous labels. To address this, we adapted RMIG (Carbonneau et al., 2022) for continuous labels and used DCI metrics from (Do & Tran, 2021). Our evaluation, including DCI, RMIG. The β-VAE and FactorVAE scores, can be found in Appendix. B.6. However, these measures suffer from limitations with sequential data and do not provide measures of attribute alignment under ground truth variation. To overcome this, consider cross-correlation between latent variables z_m (latent anchor) and its augmentation z_m^+ with respect to ground truth attribute y_m. Yet, practical challenges arise with multiple attributes, as this measure can be sensitive to variations within the same attribute. To address this, we introduce the compact **T**ime **D**isentanglement **S**core.

Metric	Align-axis	Unbiased	No. Condition
β-VAE			
FactorVAE	✓		
RMIG	✓		✓
SAP	✓		✓
DCI	✓	✓	
TDS (Ours)	✓	✓	✓

Table 1: TDS Compared to SOTA metric. (Red row the worst, Blue the best).

$$TDS = \frac{1}{dim(\mathbf{z})}\sum_{n\neq m}\sum_{k}\frac{||z_m - z_{n,k}^+||^2}{\text{Var}[z_m]}, \tag{6}$$

where $z_{n,k}^+$ is an augmentation of z_m, and $\text{Var}[z_m]$ variance of z_m. When a set of latent variables are not axis-aligned, each variable can contain a decent amount of information regarding two or more attributes. A wide gap between unaligned variables indicates an entanglement. TDS excels in axis alignment (c.f. Table. 1), is unbiased across hyperparameters.

5 Experiments

5.1 Architecture Settings And Framework Data Augmentation

Residual Blocs. We enhance our Residual model by replacing traditional components in residual blocks with Sigmoid Linear Units (SiLU) (Elfwing et al., 2017). SiLU offers advantages such as faster training, robust feature learning, and superior performance compared to weight normalization.

Our framework is given in Fig. 3, we set $L = 16$, and we fix an time window input to 256 steps, for the latent space dimension we fix $d_z = 16$.

Squeeze-and-Excitation on Spatial and Temporal: The SE block improves neural networks by selectively emphasizing important features and suppressing less relevant ones. Extending SE for time series data enhances the capture of significant temporal patterns in sequences.

Diosc ($L = 8$)	**KL** ↓	**RMSE** ↓	**Time (s)** ↓
ReLU	0.734	0.734	28800
SiLU	**0.671**	**0.671**	**21600**
ReLU+SE	0.721	0.721	32760
SiLU+SE	**0.582**	**0.582**	**23040**

Table 2: Metrics on UK-Dale (↓ lower is better [**Top-1**,**Top-2**]) for Diosc Variants activation function and SE, for $L = 8$.

Pipeline Augmentation for Electric Load Monitoring. Four augmentations were sequentially applied to all contrastive methods' pipeline branches. The parameters from the random search are: 1) **Crop and delay:** applied with a 0.5 probability and a minimum size of 50% of the initial sequence. 2) **Cutout or Masking:** time cutout of 5 steps with a 0.8 probability. 3) **Channel Masks powers:** each power (reactive, active, and apparent) is randomly masked out with a 0.4 probability. 4) **Gaussian noise:** random Gaussian noise is added to window activation y_m and $\mathbf{x}$ with a standard deviation of 0.1. The impact of each increase is detailed in the Appendix. B.2.

Pipeline Correlated Sampling Attributes. We evaluate the model's robustness to data correlations by examining various pairs, primarily focusing on linear correlations between two appliances and scenarios where one device correlates with two others. For this, we parameterize these correlations by sampling from a common distribution $p(y_1, y_2) \propto \exp\left(-||y_1 - \alpha y_2||^2/2\sigma^2\right)$, where α is a scaling factor, and σ indicates the strength of the correlation. We extends the (Träuble et al., 2021) framework beyond time series and adapts it to cover correlations between multiple devices operating in a T time window. Scenario examples include: **No Correlation** ($\sigma = \infty$); **Pair: 1** (clothes dryer/oven, $\sigma = 0.3$); **Pair: 2** (washing machine/dishwasher, $\sigma = 0.4$), and **Random pair** (randomly selected pairs, $\sigma = 0.8$). Additional correlation pairs are detailed in Appendix. B.3.

5.2 Experimental Setup

Datasets. We conducted experiments on three public datasets: UK-DALE (Kelly & Knottenbelt, 2015), REDD (Kolter & Johnson, 2011), and REFIT (Murray et al., 2017) providing power measurements from multiple homes. We focus on six appliances: Washing Machine, Oven, Dishwasher, Cloth Dryer, Fridge. We performed cross-tests on different dataset scenarios, each with varying sample sizes. Specifically, scenario **A** involved training on REFIT and testing on UK-DALE, $18.3k$ samples with time window $T = 256$, and frequencey of 60Hz, the test set consisted of $3.5k$ samples, scenario **B** involved training on UK-DALE and testing on REFIT with $13.3k$ samples, and scenario **C** involved training on REFIT and testing on REDD with $9.3k$ samples. The augmentation pipeline is applied for all scenarios. For training and testing under correlation, we use the corresponding sampling.

Baseline and Evaluation. We compare DIoSCwith downstream task models in energy, Bert4NILM (Yue et al., 2020) as a baseline and S2P (Yang et al., 2021b), S2S (Chen et al., 2018a), for those model we keep the same configuration as the original implémentation. We provide also variant β-TC/Factor/-VAE implemented for time series, compared to D3VAE (Li et al., 2023) and NVAE (Vahdat & Kautz, 2021), and RNN-VAE (Chung et al., 2015). We compare these model using RMSE, and we compute disentanglement metrics: RMIG, DCI, TDS. The metrics have been evaluated by both, either sampling from the correlated data or from the uncorrelated distribution.

Experimental Platform. We conduct 5 seeds of experiments, reporting the averaged results and standard deviation. Based on the grid search, we found that Diosc's best performance is obtained by ($\lambda = 2.3$, $\beta = 1.5$). The experiments are performed on four NVIDIA A100 GPUs. Hyperparameter settings are available in Appendix B.

5.3 Performance and Informativity of Contrastive

Finding: *DIoSC performs better in Out-of-distribution (under correlated data).*

To assess DIoSC's regularization robustness to correlated appliances, we examine scenarios involving pairs defined in §5.1. From Fig. 6.1 the increased disentanglement through Diosc gives consistent improvements in all cases, and gets more pronounced in the low data regime, indicating higher sample efficiency, as expected from better disentanglement even the correlated in pairs. Fig. 3 show the regression results as we see even when signals are correlated the disentangling is acheived and

relative improvements up to +20% in RMSE. We find factorization of supports using Diosc on the training data is strongly relate to downstream disentanglement even when experiencing strong correlation during training.

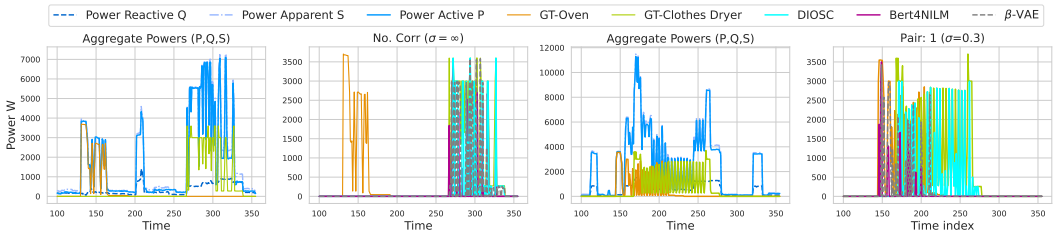

Figure 4: Prediction Clothes dryer under in correlated case (left) and uncorrelated case (right) over a time window of 256min. Moving from left to right, the graph illustrates the aggregated power (**P**,**Q**,**S**) alongside the ground-truth (**Green**) to be identified under correlation with Oven (**Orange**).

Sc.	Methods	No Corr $\sigma = \infty$			Pairs: 1 $\sigma = 0.3$			Pairs: 2 $\sigma = 0.4$			Random Pair $\sigma = 0.8$		
	Metrics –>	DCI ↓	TDS ↓	RMSE ↓	DCI ↓	TDS ↓	RMSE ↓	DCI ↓	TDS ↓	RMSE ↓	DCI ↓	TDS ↓	RMSE ↓
A	Bert4NILM	-	-	56.4 ±2.58	-	-	70.2±1.45	-	-	72.08±0.96	-	-	70.92±1.15
	S2S	-	-	54.3 ±3.12	-	-	69.5±3.56	-	-	72.31±2.45	-	-	69.95±3.26
	β-VAE	72.4±3.10	0.96±.15	48.6 ±2.32	72.4±3.10	0.96±.15	52.6 ±2.31	72.4±3.10	0.96±.15	54.73±1.54	74.29±2.04	1.08±.09	52.99±1.91
	β-TCVAE	78.0±1.09	0.94±.13	43.2 ±2.23	78.0±1.09	0.94±.13	49.2 ±1.13	77.23±0.76	0.94±.13	50.87±1.17	79.74±0.84	1.07±.11	49.65±1.43
	FactorVAE	68.4±2.41	0.97±.03	47.7 ±1.35	68.4±2.41	0.97±.03	53.2 ±1.02	69.78±1.43	0.97±.03	54.32±0.64	69.95±1.63	1.00±.02	53.45±0.82
	HFS	79.8 ± .10	0.64 ± .05	57.2 ± 2.15	79.8 ± .10	0.64 ± .05	61.3 ± 1.82	79.56±0.28	0.64±.05	62.33±1.23	80.37±.05	0.72±.03	61.64±1.52
	β-VAE + HFS	73.1±1.01	0.69±.02	34.4±1.89	73.1±1.01	0.69±.02	38.1±1.34	73.59±0.86	0.69±.04	39.65±0.87	74.25±0.59	0.73±.05	38.48±1.04
	β-TCVAE + HFS	**67.2±2.01**	**0.52±.02**	**24.3 ±1.81**	**67.2±2.01**	**0.52±.02**	**27.4 ±1.13**	**67.51±1.84**	**0.52±.07**	**28.94±0.66**	**68.79±1.27**	**0.58±.04**	**27.77±0.83**
	Diosc	**63.5±1.35**	**0.49±.02**	**19.6±1.95**	**69.3±1.2**	**0.4±.02**	**22.3±1.79**	**70.3±0.82**	**0.49±.02**	**23.97±1.19**	**67.12±0.91**	**0.51±.01**	**22.63±1.49**
B	Bert4NILM	-	-	57.85 ±1.88	-	-	68.8±1.12	-	-	73.41±1.35	-	-	72.78±0.88
	S2S	-	-	56.38 ±2.22	-	-	67.8±2.76	-	-	73.95±1.91	-	-	70.92±2.25
	β-VAE	73.78±2.68	1.08±.09	50.14 ±1.87	75.47±1.98	0.82±.10	51.7±1.79	70.8±2.62	0.85±.11	55.98±1.27	76.18±1.54	1.16±.08	54.83±1.58
	β-TCVAE	79.57±0.84	1.07±.11	45.72 ±1.68	80.23±0.54	0.81±.09	48.3±0.94	76.2±0.54	0.83±.10	51.74±0.94	80.88±0.53	1.15±.10	51.15±1.10
	FactorVAE	70.14±1.89	1.00±.02	49.02 ±1.05	71.89±1.24	0.94±.02	52.4±0.85	68.7±1.13	0.92±.02	55.24±0.42	71.57±1.27	1.06±.01	54.68±0.64
	HFS	80.12±.05	0.72±.03	58.49 ±1.45	80.26±.03	0.56±.03	6.0±1.42	78.8±0.15	0.58±.03	63.79±0.97	80.61±.02	0.80±.02	63.22±1.17
	β-VAE + HFS	74.47±0.61	0.73±.05	36.09±1.25	75.12±0.41	0.67±.02	37.4±1.04	72.8±0.52	0.64±.03	40.92±0.66	75.07±0.43	0.75±.03	39.68±0.80
	β-TCVAE + HFS	**68.54±1.36**	**0.58±.04**	**25.88 ±1.20**	**69.28±1.01**	**0.46±.01**	**26.7±0.88**	**66.7±1.51**	**0.45±.02**	**29.82±0.51**	**7.04±0.93**	**0.72±.02**	**40.49±0.64**
	Diosc	**64.42±0.96**	**0.51±.01**	**21.35 ±1.80**	**65.11±0.66**	**0.39±.01**	**21.5±1.44**	**69.5±0.43**	**0.48±.01**	**24.94±0.87**	**65.05±0.71**	**0.55±.01**	**24.05±1.30**
C	Bert4NILM	-	-	58.29 ±2.16	-	-	75.6±1.68	-	-	71.73±1.66	-	-	76.05±1.87
	S2S	-	-	56.28 ±2.43	-	-	73.8±3.91	-	-	74.76±3.75	-	-	73.47±4.12
	β-VAE	74.17±2.01	1.03±.09	50.18 ±1.92	73.84±1.56	0.72±.12	55.7±2.47	76.1±3.36	1.07±.17	56.32±2.31	73.95±1.93	1.16±0.11	55.90±2.40
	β-TCVAE	79.21±0.89	0.98±.10	45.11 ±2.03	79.48±0.75	0.78±.08	50.9±1.27	78.85±0.94	1.05±.15	51.19±1.84	80.57±0.95	1.10±0.11	51.17±1.85
	FactorVAE	70.23±1.70	0.99±.02	49.12 ±1.18	69.75±1.53	0.99±.03	56.4±1.11	70.92±1.58	0.99±.05	55.48±1.25	70.43±1.74	1.05±.02	54.61±1.34
	HFS	8.04±.06	0.67 ± .03	59.04 ±1.74	80.11±.05	0.60±.04	62.9±1.98	79.91±0.36	0.69±.07	63.52±1.94	80.42±.06	0.73±.03	63.83±2.01
	β-VAE + HFS	74.03±0.79	0.70±.01	35.65±1.59	74.14±0.82	0.74±.01	40.5±1.49	74.26±0.95	0.71±.06	40.32±1.38	74.84±0.51	0.78±.05	39.38±1.19
	β-TCVAE + HFS	**69.04±1.45**	**0.54±.01**	**25.85 ±1.45**	**68.37±1.31**	**0.47±.01**	**28.9±1.28**	**69.07±2.02**	**0.59±.09**	**30.38±1.24**	**69.84±1.43**	**0.62±.04**	**29.29±1.13**
	Diosc	**64.87±1.07**	**0.50±.01**	**19.6±1.95**	**70.54±0.60**	**0.50±.01**	**21.1±1.92**	**71.2±0.94**	**0.44±.03**	**26.97±1.04**	**67.72±1.01**	**0.57±0.01**	**24.12±1.58**

Table 3: Average scores DCI, TDS, and RMSE vary from No Correlation (left) to every appliance correlated with one confounder (right) on uncorrelated test data. Red to blue, with bold indicating the best performance per correlation. (↓ lower is better, ↑ higher is better [**Top-1**,**Top-2**]).

6 Ablation Studies

6.1 Diosc preserves its robustness in correlated scenarios

Finding: *Diosc preserves its robustness in correlated scenarios and achieves comparable performance to baseline models with less training sample data.*

Training with the same l variational inference model with the different regularisation variants results shows that Diosc outperforms SOTA as shown in Fig. 5, mainly in the uncorrelated cases with only 50% of labelled data, which corresponds to HDF (Roth et al., 2023).

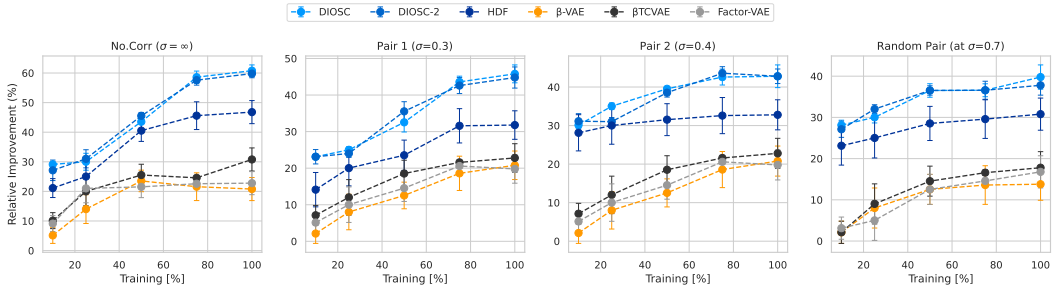

Figure 5: Relative RMSE (%) improvement over Bert4NILM for six devices using Diosc, β-VAE, and FactorVAE, with the amount of labelled training data as a variable parameter.

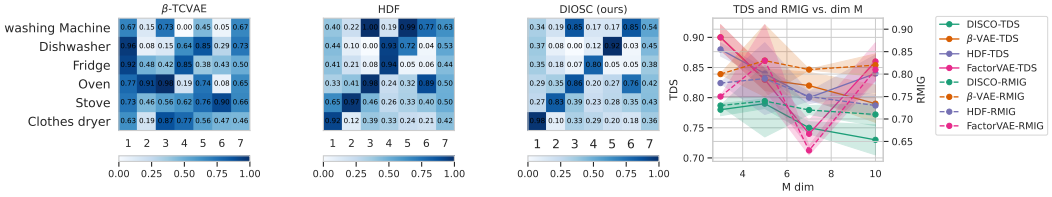

Figure 6: We find strong correlation between Diosc and disentanglement metrics vary with M (right), linked to classifcation accuracy of each compenents z_m on y_m labeled test data (Left), Darker Blue $\rightarrow$ High accuracy.

With 80% of data, Diosc scores 14% better than HDF and 61.4% better than the baseline Bert4NILM. In the correlated scenarios (pair 1 and 2), β/Factor/TC-VAE shows weaker performance, while Diosc consistently outperforms HDF and the baseline.

6.2 In-depth self-attention l-VAEs learn an effective representation.

Finding: *As Diosc deepens, representation increases over 20% (40% in TDS), downtasking boosts performance, and attention mechanisms contribute a 10% improvement.*

Method	Depth (L)	NRMSE ↓	RMIG ↓	TDS ↓
VAE (baseline)	-	0.928	0.921	0.935
VAE (baseline)+Diosc	-	0.929	0.924	0.931
FactorVAE	-	0.942	0.931	0.973
β-TCVAE	-	0.931	0.918	0.937
β-TCVAE+Diosc	-	0.930	0.922	0.933
DIP-VAE	-	0.932	0.915	0.939
DIP-VAE+Diosc	-	0.928	0.926	0.930
Diosc	8	0.50	0.73	0.71
Diosc w/o Attention	8	0.54	**0.71**	0.72
Diosc	16	**0.49**	0.74	0.70
Diosc w/o Attention	16	0.52	**0.72**	0.73
Diosc	32	**0.48**	0.75	**0.69**

Table 4: Average Normalized RMSE, RMIG, and TDS Scores for Variants Diosc w/,w/o Attention, as L Increases. (↓ lower values are better [**Top-1**,**Top-2**], the Red row the worst on average, and the Blue the best).

In Table 7, we employ l-Variational Inference with the Diosc regularizer, both with and without self-attention, and explore its application with alternative structures tailored for time series, particularly those residual in D3VAE. Our observations reveal two key findings. Firstly, incorporating Diosc with another regularization method slightly enhances results, as the alternative regularizer assumes independent factorization, potentially compromising the relaxing effect. Secondly, Diosc demonstrates improved performance with increasing values of L, and the TDS correlates positively with performance, while RMIG suggests that using Diosc with attention leads to well-disentangled representations. Notably, the attention mechanism proves efficient by enhancing the overall model performance.

6.3 Robustness, Disentanglement, and Strong Generalization

Finding: *Diosc demonstrates robust disentanglement performance across varying dimensions, while FactorVAE exhibits degradation as dimensionality increases $M \uparrow$.*

In Fig. 6 (right), we report the disentanglement performance of Diosc and FactorVAE on the Ukdale dataset as M is increased. FactorVAE (Higgins et al., 2016) is the closest TC-based method it uses a single monolithic Discriminator and the density-ratio trick to explicitly approximate TC. Computing $TC(\mathbf{z})$ is challenging to compute as M increases. The results for $M = 10$ (scalable $\approx \times 3$) are included for comparison. The average disentanglement scores for Diosc $M = 7$ and $M = 10$ are lower and very close, indicating that its performance is robust in M. This is not the case for HDF Factor/β-VAE. It performs worse on all metrics when m increases. Interestingly, HDF $M = 7$ seems to recover its performance on most metrics. Despite this, the difference suggests that HDF and Factor/β-VAE are not robust to changes in M. The optimal M for HDF and TC/β-VAE, shown in Fig. 6 (left), indicates promising accuracy for HDF, despite being no better than Diosc.

7 Conclusion and Future Work

To overcome the limitation of assuming independence in existing time series disentanglement methods, which doesn't align with real-world correlated data, our approach focuses on recovering correlated data. By weakly supervising the independence factorization assumption to independence-of-support, our method achieves disentanglement by enabling the model to encode attribute variability in the latent space. Using Diosc, a combination of contrastive regularization and l-Variational Autoencoder for time series, we show that promoting pairwise factorized support suffices for disentangling time series. Diosc excels in downstream tasks of NILM, showing over **+61.4%** relative improvements regarding baseline across datasets with various correlation shifts. Enhanced disentanglement aids out-of-distribution generalization in representation learning. Future work may explore support factorization for time series with causal notions.

REFERENCES

Ifigeneia Apostolopoulou, Ian Char, Elan Rosenfeld, and Artur Dubrawski. Deep attentive variational inference. In *International Conference on Learning Representations*, 2021.

Robert B Ash. *Information theory*. Courier Corporation, 2012.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives, April 2014. URL http://arxiv.org/abs/1206.5538. arXiv:1206.5538 [cs].

Giovanni Bucci, Edoardo Fiorucci, Simone Mari, and Andrea Fioravanti. A New Convolutional Neural Network-Based System for NILM Applications. *IEEE Transactions on Instrumentation and Measurement*, 2021. doi: 10.1109/TIM.2020.3035193.

Marc-André Carbonneau, Julian Zaidi, Jonathan Boilard, and Ghyslain Gagnon. Measuring Disentanglement: A Review of Metrics, May 2022. URL http://arxiv.org/abs/2012.09276. arXiv:2012.09276 [cs].

Kunjin Chen, Qin Wang, Ziyu He, Kunlong Chen, Jun Hu, and Jinliang He. Convolutional sequence to sequence non-intrusive load monitoring. *the Journal of Engineering*, 2018(17):1860–1864, 2018a. Publisher: Wiley Online Library.

Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating Sources of Disentanglement in Variational Autoencoders. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018b.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A Recurrent Latent Variable Model for Sequential Data. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/b618c3210e934362ac261db280128c22-Abstract.html.

Kien Do and Truyen Tran. Theory and Evaluation Metrics for Learning Disentangled Representations, March 2021. URL http://arxiv.org/abs/1908.09961. arXiv:1908.09961 [cs, stat].

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning, November 2017. URL http://arxiv.org/abs/1702.03118. arXiv:1702.03118 [cs].

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016.

Jack Kelly and William Knottenbelt. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific data*, 2, 2015. Publisher: Nature Publishing Group.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning, March 2021. URL http://arxiv.org/abs/2004.11362. arXiv:2004.11362 [cs, stat].

Hyunjik Kim and Andriy Mnih. Disentangling by Factorising, July 2019. URL http://arxiv.org/abs/1802.05983. arXiv:1802.05983 [cs, stat].

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

David Klindt, Lukas Schott, Yash Sharma, Ivan Ustyuzhaninov, Wieland Brendel, Matthias Bethge, and Dylan Paiton. Towards nonlinear disentanglement in natural data with temporal sparse coding. *arXiv preprint arXiv:2007.10930*, 2020.

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

Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative Time Series Forecasting with Diffusion, Denoise, and Disentanglement, January 2023. URL http://arxiv.org/abs/2301.03028. arXiv:2301.03028 [cs].

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling, November 2019. URL http://arxiv.org/abs/1902.02102. arXiv:1902.02102 [cs, stat].

David Murray, Lina Stankovic, and Vladimir Stankovic. An electrical load measurements dataset of united kingdom households from a two-year longitudinal study. *Scientific data*, 4(1):1–12, 2017.

Christoforos Nalmpantis and Dimitris Vrakas. On time series representations for multi-label NILM. *Neural Computing and Applications*, 32(23), 2020. ISSN 0941-0643. doi: 10.1007/s00521-020-04916-5. URL https://link.springer.com/epdf/10.1007/s00521-020-04916-5.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. *arXiv preprint arXiv:2102.10543*, 2021.

Karsten Roth, Mark Ibrahim, Zeynep Akata, Pascal Vincent, and Diane Bouchacourt. Disentanglement of Correlated Factors via Hausdorff Factorized Support, February 2023. URL http://arxiv.org/abs/2210.07347. arXiv:2210.07347 [cs, stat].

Ramalingam Shanmugam. Elements of causal inference: foundations and learning algorithms. *Journal of Statistical Computation and Simulation*, 88(16):3248–3248, November 2018. ISSN 0094-9655, 1563-5163. doi: 10.1080/00949655.2018.1505197. URL https://www.tandfonline.com/doi/full/10.1080/00949655.2018.1505197.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On disentangled representations learned from correlated data. In *International Conference on Machine Learning*, pp. 10401–10412. PMLR, 2021.

Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, 2020.

Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder, January 2021. URL http://arxiv.org/abs/2007.03898. arXiv:2007.03898 [cs, stat].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yixin Wang and Michael I. Jordan. Desiderata for Representation Learning: A Causal Perspective, February 2022a. URL http://arxiv.org/abs/2109.03795. arXiv:2109.03795 [cs, stat].

Yixin Wang and Michael I. Jordan. Desiderata for Representation Learning: A Causal Perspective, February 2022b. URL http://arxiv.org/abs/2109.03795. arXiv:2109.03795 [cs, stat].

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. *arXiv preprint arXiv:2202.01575*, 2022.

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9588–9597, Nashville, TN, USA, June 2021a. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.00947. URL https://ieeexplore.ieee.org/document/9578520/.

Mingzhi Yang, Xinchun Li, and Yue Liu. Sequence to Point Learning Based on an Attention Neural Network for Nonintrusive Load Decomposition. *Electronics*, 2021b.

Yandong Yang, Jing Zhong, Wei Li, T. Aaron Gulliver, and Shufang Li. Semisupervised Multi-label Deep Learning Based Nonintrusive Load Monitoring in Smart Grids. *IEEE Transactions on Industrial Informatics*, 16(11):6892–6902, November 2020. ISSN 1551-3203, 1941-0050. doi: 10.1109/TII.2019.2955470. URL https://ieeexplore.ieee.org/document/8911216/.

Weiran Yao, Guangyi Chen, and Kun Zhang. Temporally disentangled representation learning. *Advances in Neural Information Processing Systems*, 35:26492–26503, 2022.

Zhenrui Yue, Camilo Requena Witzig, Daniel Jorde, and Hans-Arno Jacobsen. BERT4NILM: A Bidirectional Transformer Model for Non-Intrusive Load Monitoring. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*, NILM'20, pp. 89–93, New York, NY, USA, November 2020. Association for Computing Machinery. ISBN 978-1-4503-8191-8. doi: 10.1145/3427771.3429390. URL https://dl.acm.org/doi/10.1145/3427771.3429390.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction, June 2021. URL http://arxiv.org/abs/2103.03230. arXiv:2103.03230 [cs, q-bio].

Wentian Zhao, Yanyun Gao, Tingxiang Ji, Xili Wan, Feng Ye, and Guangwei Bai. Deep temporal convolutional networks for short-term traffic flow forecasting. *Ieee Access*, 7:114496–114507, 2019.

Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive Learning Inverts the Data Generating Process, April 2022. URL http://arxiv.org/abs/2102.08850. arXiv:2102.08850 [cs].

SUPPLEMENTARY MATERIAL

In order to facilitate a comprehensive examination of our paper, we present additional results and furnish complete proofs for the assumptions articulated in the main manuscript. This supplementary material is meticulously organized as follows:

- Appendix. A: l-Variational Autoencoder Procedure (Inference and Generative)
  - Appendix. A: Algorithm Inference q_ϕ
  - Appendix. A: Algorithm Generative p_θ
- Appendix. B: Implementation Details and Extended Ablation Studies 5.1
  - Appendix. B.1: Implementations Details.
  - Appendix. B.3: Augmentation and Sampling correlated Appliances.
  - Appendix. B.7.1: Ablation on using multiple autoencoders for training, each dedicated to a specific latent variables.
- Appendix. C: Connection between DIoSC, Causality-Representation and the Information Bottleneck Principle

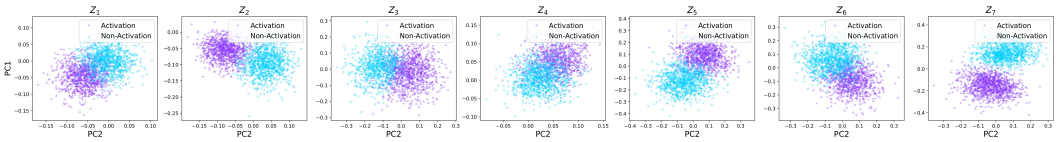

Figure 7: PCA visualization on the latent variable of DIoSC. A distinct separation between samples corresponds to the activation and inactivation of an appliance. The latent space exhibits clear distinguishability between instances of activation and non-activation of each appliance from the lefet to right: Washing Machine, Oven, Dishwasher, Cloth Dryer, and Fridge. (**Cyan** represents non-activated samples, while **blue-purple** indicates the activation of appliances in those samples).

A l-VARIATIONAL AUTOENCODER PROCEDURE (INFERENCE AND GENERATIVE)

To avoid time locality during dimension reduction, and keep long-range capability we refer to an in-depth Temporal Attention with l-Variational layers. Unlike NVAE (Vahdat & Kautz, 2020) for which the latent space $\mathbf{z}$ is level-structured locally, in this work, we enable the model to establish strong couplings, as depicted. The core problem we aim to address is to construct a feature $\hat{T}^{(l)}$ (Time context) that effectively captures the most informative features from a given sequence $T^{<l} = \{T^i\}_{i=1}^{(l)}$. Both $\hat{T}^{(l)}$ and $T^{(l)}$ are features with the same dimensionality: $\hat{T}^{(l)} \in \mathbb{R}^{T\times C}$ and $T^i \in \mathbb{R}^{T\times C}$. In our model, we employ Temporal Attention to construct either the prior or posterior beliefs of variational layers, which enables us to handle long context sequences with large dimensions T effectively. The construction of $\hat{T}^{(l)}$ relies on a query feature $\mathbf{Q}^{(l)} \in \mathbb{R}^{T\times Q}$ of dimensionality Q with $Q \ll C$, and the corresponding context $T^{(l)}$ is represented by a key feature $\mathbf{K}^{(l)} \in \mathbb{R}^{T\times Q}$. Importantly, $\hat{T}^{(l)}(t)$ of time step i in sequence T depends solely on the time instances in $T^{<l}$.

$$\hat{T}^{(l)}(t) = \sum_{i<l} \alpha_{i\to l}(t) \cdot T^{(l)}(t) \tag{7}$$

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 $\hat{T} = \mathbf{A}(T^{<l}, \mathbf{Q}^{(l)}, \mathbf{K}^{(<l)})$.

A powerful extension to the above single attention mechanism is the multi-head attention introduced in (Vaswani et al., 2017), which allows the model to jointly attend to information from different representation subspaces at different scales. Instead of computing a single attention function, this

method first projects Q, K, V onto h different vectors, respectively. An attention function $\mathbf{A}(\cdot)$ is applied individually to these h projections. The output is a linear transformation of the concatenation of all attention outputs:

$$\text{Multi-}\mathbf{A}(Q,K,V) = \oplus\{\mathbf{A}(QW_{qi}, KW_{ki}, VW_{vi})\}_{i=1}^{h} W_o, \tag{8}$$

Where W_o, W_{qi}, W_{ki}, W_{vi} are learnable parameters of some linear layers. $QW_{qi} \in \mathbb{R}^{n_q \times d_{hq}}$, $KW_{ki} \in \mathbb{R}^{n_v \times d_{hk}}$, $VW_{vi} \in \mathbb{R}^{n_v \times d_{hv}}$ are vectors projected from Q, K, V respectively. $d_{hq} = \frac{d_q}{h}$ and $d_{hv} = \frac{d_v}{h}$. Following the architecture of the transformer (Vaswani et al., 2017), we define the following multi-head attention block:

$$Q_0 = \text{LayerNorm}(\oplus\{QW_{q1}\}_{i=1}^{h} + \text{MultiAtt}(Q,K,V)), \tag{9}$$

$$\text{MultiBloc-}\mathbf{A}(Q,K,V) = \text{LayerNorm}(Q_0 + Q_0 W_{q0}), \tag{10}$$

where $W_{q0} \in \mathbb{R}^{d_q \times d_q}$ is a learnable linear layer.

A.1 Encodeur (Infrence Model q_ϕ)

As shown in Fig 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 Fig 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 (KL) in Eq 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:

$$\hat{T}_q^{(l)} \leftarrow \mathbf{A}(h^{\geq l}, \mathbf{Q}_q^{(l)}, \mathbf{K}_q^{\geq l}) \text{ for } l = L, L-1, ..., 1. \tag{11}$$

We adopt the Gaussian residual parametrization between the prior and the posterior. The prior is given by $p(\mathbf{z}_l|\mathbf{z}^{(<l)}) = \mathcal{N}(\mu(T_p^l, \theta), \sigma(T_p^l, \theta)$. The posterior is then given by $q(\mathbf{z}_l|\mathbf{x}, \mathbf{z}^{(<l)}) = \mathcal{N}(\mu(T_p^l, \theta) + \mathbf{\Delta}\mu(\hat{T}_q^l, \phi), \sigma(T_p^l, \theta) \cdot \mathbf{\Delta}\sigma(\hat{T}_q^l, \phi))$ where the sum $(+)$ and product $(\cdot)$ are pointwise, and T_q^l is defined in Eq 11. $\mu(\cdot)$, $\sigma(\cdot)$, $\mathbf{\Delta}\mu(\cdot)$, and $\mathbf{\Delta}\sigma(\cdot)$ are transformations implemented as convolutions layers. Based on this, For $\mathcal{L}_{KL}$ in Eq 2, the last term is approximated by: $0.5 \times \left(\frac{\mathbf{\Delta}\mu^{(l)2}}{\sigma^{(l)2}} + \mathbf{\Delta}\sigma^{(l)2} - \log \mathbf{\Delta}\sigma^{(l)2} - 1\right)$.

A.2 Decodeur (Generative Model p_θ).

The conditioning factor of the prior distribution at variational layer l is represented by context feature $T_p^{(l)} \in \mathbb{R}^{T \times C}$. A convolution is applied on $T_p^{(l)}$ to obtain parameters θ defining the prior. $\mathbf{Res}_p^{(l)}$ is a non-linear transformation of the immediately previous latent information $\mathbf{z}^{(l)}$ and prior context $T_p^{(l)}$ containing latent information from distant layers $\mathbf{z}^{(<l)l}$, such that $T_p^{(l)} = \mathbf{Res}_p^{(l)}(\mathbf{z}^{(l)} \oplus T_p^{(l)})$. $\mathbf{Res}_p^{(l)}(\cdot)$ is a transformation operation, typically implemented as a cascade of residual cells and corresponds to the blue residual module. $\mathbf{z}^{(l)}$ and $T_p^{(l)}$ are passed in from the previous layer. Because of the architecture's locality, the influence of $\mathbf{z}^{(l)}$ could potentially overshadow the signal coming from $T_p^{(l)}$. To prevent this, we adopt direct connections between each pair of stochastic layers. 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. During training, the temporal context T_p, $\mathbf{Q}_p$, and $\mathbf{K}_p$ are jointly learned:

$$[T_p^{(l)}, \mathbf{Q}_p^{(l)}, \mathbf{K}_p^{(l)}] \leftarrow \mathbf{Res}_p^{(l)}(\mathbf{z}^{(l)} \oplus (T_p^{(l)} + \eta_p^{(l)} \mathbf{A}(T_p^{<l}, \mathbf{Q}_p^{(l)}, \mathbf{K}_p^{<l}))) \text{ for } l = L, L-1, ..., 1. \tag{12}$$

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 $T_p^i \in \mathbb{R}^{T \times C}$, $\mathbf{Q}_p^{(l)} \in \mathbb{R}^{T \times Q}$, $\mathbf{K}_p^{<l} = \{\mathbf{K}_p^i\}_{i=1}^{(l)}$ with $\mathbf{K}_p^i \in \mathbb{R}^{\times Q}$, and $Q \ll C$. 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.

Algorithm 1 Decoder of l-Variational Auto-encoder

Require: Inputs
Require: $h = \{h^{(l)}\}_{l=1}^{L}$: The hidden features.
Require: $Q_q = \{k_{lq}\}_{l=1}^{L}$: The data keys.
Ensure: Initialization $T_p^0 \equiv 0$, $Q_p^1 \equiv 0$, $Q_q^{(l)} \equiv 0$, and $K_p^{(l)} \equiv 0$.
  while $l \geq L$ **do**
    $[Q_q^{(l)}, T_p^{(l)}, Q_p^{(l)}, K_p^{(l)}] \leftarrow T^{(l)}(\mathbf{z}^{(l-1)} \oplus T_{pl-1})$.
    $T_p^{(l)} \leftarrow T_p^{(l)} \oplus Q_q^{(l)}$. ▷ Used to build the prior distribution.
    if $l > 1$ **then**
      $T_p \leftarrow \text{LayerNorm}(T_{pl-1})$.
      $\hat{T}^{(l)} \leftarrow \text{MultiHead-SelfAttention}(\{T_{l'}\}_{l'=1}^{l}, Q^{(l)}, \{k_{l'}\}_{l'=1}^{l})$.
      $T_p^{(l)} \leftarrow T_p^{(l)} + \gamma^{(l)}(\hat{T}^{(l)} + \text{LayerNorm}(\hat{T}^{(l)}))$.
    else
      $T_p^{(l)} \leftarrow T^{(l)}$.
    end if
    $[\mu^{(l)}, \sigma^{(l)}] \leftarrow \text{Linear}^{(l)}(T_p^{(l)})$. ▷ Compute mean and variance
    $p(\mathbf{z}_l|\mathbf{z}^{(<l)}) = \mathcal{N}(\mu_p^{(l)}, \sigma_p^{(l)})$. ▷ Used to form and sample from the posterior distribution.
    if $l < L$ **then**
      $[T^{(l)}, k^{(l)}] \leftarrow h^{(l)} \oplus T_p^{(l)}$.
      $\hat{T}^{(l)} \leftarrow \text{MultiHead-SelfAttention}(\{h_{l'}\}_{l'=l}^{L} \cup \{T_{l'}, k_{l'}\}_{l'=l}^{L}, Q^{(l)}, \{k_{l'}\}_{l'=l}^{L})$.
      $T_{ql} \leftarrow T_{ql} + \text{LayerNorm}(T_{ql})$.
      $T_p^{(l)} \leftarrow T_p^{(l)} + \gamma^{(l)}(\hat{T}^{(l)} + \text{LayerNorm}(\hat{T}^{(l)}))$.
    else
      $T_p^{(l)} \leftarrow h^{(l)} \oplus T_p^{(l)}$.
    end if
    $[\Delta\mu^{(l)}, \Delta\sigma^{(l)}] \leftarrow \text{Linear}^{(l)}(T_p^{(l)})$. ▷ Compute mean and variance
    $q(\mathbf{z}_l|x, \mathbf{z}^{(<l)}) = \mathcal{N}(\mu_p^{(l)} + \Delta\mu_q^{(l)}, \sigma_p^{(l)}\Delta\sigma^{(l)}q)$. ▷ Parameterize Residual
    $\mathbf{z}_l \sim q(\mathbf{z}_l|\mathbf{x}, \mathbf{z}^{(<l)})$. ▷ Sample latent variables
  end while
  Return:
  $T_p = \{T_p^{(l)}\}_{l=1}^{L}$: Latent context features.
  $z = \{\mathbf{z}_l\}_{l=1}^{L}$: Inferred latent variables.
  $q(z|x) = \prod_{l=1}^{L} q(\mathbf{z}_l|x, \mathbf{z}^{(<l)})$: Approximate posterior distribution of $\mathbf{z}$.
  $p(z) = p(\mathbf{z}_l|\mathbf{z}^{(<l)})$: Prior distribution of $\mathbf{z}$.

B Implementation Details and Extended Ablation Studies

B.1 Data sets

In this section we expand further on the datasets we performed experiments on. Our experiement are conducted experiments on three public datasets: UK-DALE (Kelly & Knottenbelt, 2015), REDD (Kolter & Johnson, 2011), and REFIT (Murray et al., 2017) providing power measurements from multiple homes (5 house for Uk-Dale, 6 for REFIT and 5 for REDD). Our focus was on six appliances: Washing Machine, Oven, Dishwasher, Cloth Dryer, Fridge. We performed cross-tests on different dataset scenarios, each with varying sample sizes. Specifically, scenario **A** involved training on REFIT and testing on UK-DALE, $18.3k$ samples with time window $T = 256$, and frequency of 60Hz, the test set consisted of $3.5k$ samples, scenario **B** involved training on UK-DALE and testing on REFIT with $13.3k$ samples, and scenario **C** involved training on REFIT and testing

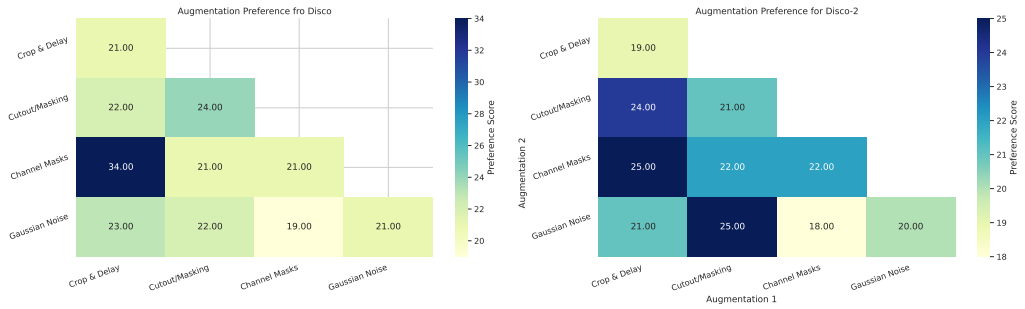

Figure 8: We assess the efficacy of augmentations on both Diosc and Diosc-2 models, and it is evident that Gaussian noise enhances performance while decreasing the influence of channel power and delay. This observation aligns with real-world scenarios, where a sensor's data transmission delay introduces a lag between the aggregated source and the power consumption of individual appliances. The collective impact of all four augmentations suggests that the contrastive properties regarding appliance profiles are adequately captured during training.

on REDD with $9.3k$ samples. The augmentation pipeline is applied for all scenarios. For training and testing under correlation, we use the corresponding sampling.

Sc.	Data	Train	Test	Val
A	Number of Activation devices	29 250	6 281	6 371
	Decompensation Samples	18977	3 501	4338
	Number of Positives	11 260	2 383	2752
B	Number of Activation devices	10 250	2 281	6 371
	Decompensation Samples	18904	3513	2338
	Number of Positives	9260	9 383	2752
C	Number of Activation devices	9 132	3 181	1 371
	Decompensation Samples	13977	9310	3338
	Number of Positives	10160	5 383	1214

Table 5: Number of activation of device and Samples for Different Data Sets and Tasks

B.2 Pipeline Augmentation for Electric Load Monitoring.

Four augmentations were sequentially applied to all contrastive methods' pipeline branches. The parameters from the random search are:

1. **Crop and delay:** applied with a 0.5 probability and a minimum size of 50% of the initial sequence.
2. **Cutout or Masking:** time cutout of 5 steps with a 0.8 probability.
3. **Channel Masks powers:** each power (reactive, active, and apparent) is randomly masked out with a 0.4 probability.
4. **Gaussian noise:** random Gaussian noise is added to window activation y_m and $\mathbf{x}_m$ with a standard deviation of 0.1 (augmentation 1) and 0.3 (augmentation 2). The impact of each increase is detailed in Fig. **??** bellow.

B.3 Pipeline Correlated samples.

Robustness of the model to correlations between data is assessed by examining different pairs. We focus mainly on linear correlations between two different devices and on the case where one device correlates with two others. To do this, we parameterize the correlations by sampling a

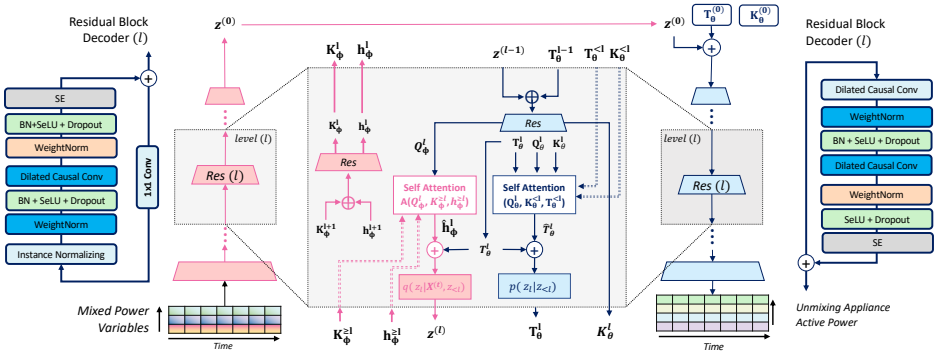

Figure 10: Architecture of Diosc with attention l-Variational Inference.

dataset from the common distribution. We build on the correlation time series framework by introducing a pairwise correlation between the attributes y_m and y_n as follows: $p(y_m, y_n) \propto \exp\left(-||y_m - \alpha y_n||^2/2\sigma^2\right)$, where α is a scaling factor. A high value of σ indicates a lower correlation between the normalised attributes y_m and y_n (No.Corr, $\sigma = \infty$). We also extend this framework to cover correlations between several attributes in the time window T. Therefore, we consider correlation pair scenarios such as : *No correlation*; *Pair:1* washer-dryer; *Pair:2* dryer-oven and, finally, a *Random pair:* approach with randomly selected appliances (see Fig. 9).

- **No Corr.:** No Correlation during training. (default evaluation setting)
- **Pair: 1** washer-dryer.
- **Pair: 2** dryer-oven.
- **Pair: 3** lighting and television.
- **Pair: 4** microwave and oven.
- **Pair: 5** washer-dishwasher.
- **Random Pairs**,

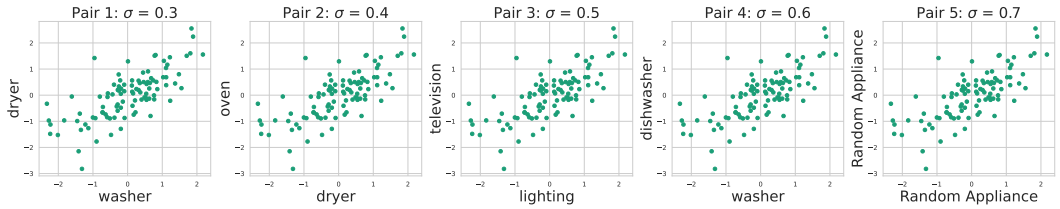

Figure 9: Correlation among paired samples at a constant value of σ.

B.4 Further Experimental Details

In this section, we expand on the details of our architecture. The full architecture of the encoder and decoder is depicted in Fig 10. Because we deal with time-series, we used causal dilated convolutions to not break temporal ordering. We implement all our experiments using the PyTorch framework (Paszke et al., 2019). For exact and fair comparability, we `https://anonymous.4open.science/r/DisCo/`.

Encoders f_ϕ and Decoders f_θ Our model in Fig. 10) use a bidirectional encoder layers, which hierarchically processes input data to generate a low-resolution latent code refined by a series of upsampling layers. The initial phase involves a rudimentary encoder that produces a low-resolution latent code. This code is then refined by a series of upsampling layers in the "Residual Decoders" blocks build on causal convolution, gradually increasing the resolution. At each step of the refinement process, the use of "Residual Encoders and Decoders" efficiently captures semantic features, while the temporal attention in the "Residual Decoders," implemented by dilated causal conv (Zhao et al., 2019), ensures the temporal dependence of z. In our architecture, the smallest dimension of $\mathbf{z}$

is set to $\mathbf{z} \in \mathbb{R}^{d_z \times (M+K)}$ with $d_z = 16$ and $M = 6$, $K = 1$, representing the number of appliances to be separated in a mixed sequence of size $T = 256$.

Augmented data: All four augmentations are employed during the training process, and we have opted to augment the data by 50% across all Scenarios A, B, and C.

Optimization In all our experiments, we used the Adam optimizer (Kingma & Ba, 2014) 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 enhance training stability and applied early stopping after 5 iterations. We set $\alpha = 0.5$ and $\beta = 2.5$ after a grid search for the best model convergence on the validation data.

Method	Parameter	Values
β-VAE	β	[0.1,0.2,0.3,0.4,0.5,0.6,0.7,1,,2.5]
β-TCVAE	β	[0.1,0.2,0.3,0.4,0.5,0.6,0.7,1,,2.5]
FactorVAE	β	[0.1,0.2,0.3,0.4,0.5,0.6,0.7,1,,2.5]
HFS	γ	[0.1,0.2,0.3,0.4,0.5,0.6,0.7,1,,2.5]
β-VAE + HFS	γ	[1,2,3,8,11,12,13,14,15,16]
DIOSC	η	[0.1,0.2,0.3,0.8,1,1.5,2,2.5,3,3.5]
DIOSC-2	η	[0.1,0.2,0.3,0.8,1,1.5,2,2.5,3,3.5]

Table 6: Hyperparameter testing for each method.

B.5 Impact of η, λ, β and Training stability

In this section, we expand on the details the satability of training and perofromance acrross values of η.

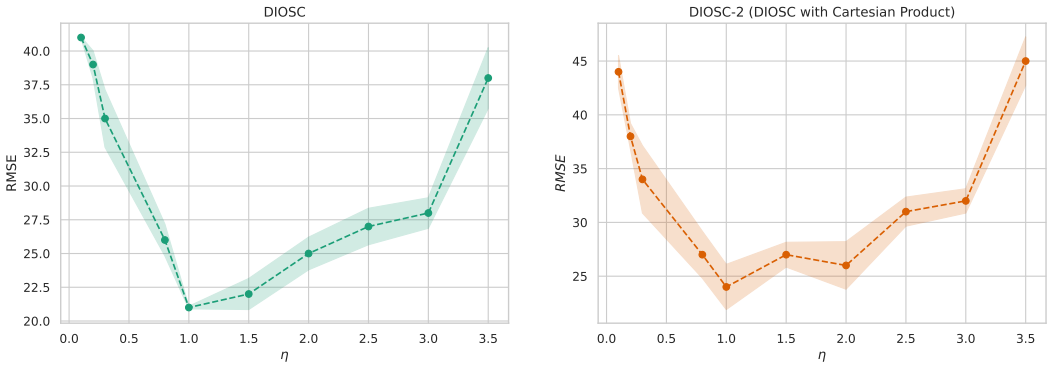

Figure 11: Root Mean Squared Error (RMSE) for various values of η.

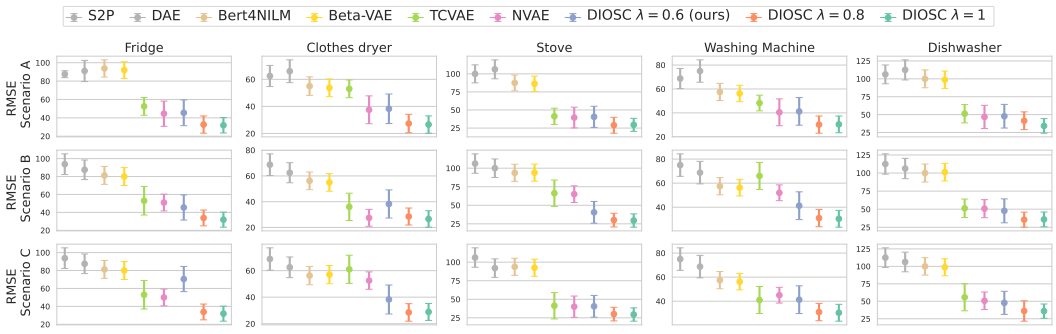

Figure 12: RMSE in $Watt^2$ across scenarios **A**, **B**, and **C**. We see the impact of λ on the reconstruction of powers of each appliances.

B.6 Implementation of Metrics and Study Case of TDS

In implementing the disentanglement metrics, we adhere to the methodology outlined in (Locatello et al., 2019), expanding it to accommodate time series data. For the computation of DCI metrics, we employ a gradient boosted tree from the scikit-learn package.

B.6.1 β-VAE Metric

(Higgins et al., 2016) suggest fixing a random attributes of variation in the underlying generative model and sampling two mini-batches of observations x. Disentanglement is then measured as the accuracy of a linear classifier that predicts the index of the fixed factor based on the coordinate-wise sum of absolute differences between the representation vectors in the two mini-batches. We sample two batches of 256 points with a random factor fixed to a randomly sampled value across the two batches, and the others varying randomly. We compute the mean representations for these points and take the absolute difference between pairs from the two batches. We then average these 64 values to form the features of a training (or testing) point.

B.6.2 FactorVAE Metric

(Kim & Mnih, 2019) address several issues with this metric by using a majority vote classifier that predicts the index of the fixed ground-truth attribute based on the index of the representation vector with the least variance. First, we estimate the variance of each latent dimension by embedding $10k$ random samples from the data set, excluding collapsed dimensions with variance smaller than .05. Second, we generate the votes for the majority vote classifier by sampling a batch of 64 points, all with a factor fixed to the same random value. Third, we compute the variance of each dimension of their latent representation and divide it by the variance of that dimension computed on the data without interventions. The training point for the majority vote classifier consists of the index of the dimension with the smallest normalized variance. We train on $10k$ points and evaluate on $5k$ points.

B.6.3 Mutual Information Gap Metric

(Chen et al., 2018b) argue that the BetaVAE metric and the FactorVAE metric are neither general nor unbiased as they depend on some hyperparameters. They compute the mutual information between each ground-truth factor and each dimension in the computed representation $r(x)$. For each ground-truth factor zk, they then consider the two dimensions in $r(x)$ that have the highest and second highest mutual information with zk. The Mutual Information Gap (MIG) is then defined as the average, normalized difference between the highest and second highest mutual information of each factor with the dimensions of the representation. The original metric was proposed evaluating the sampled representation. Instead, we consider the mean representation, in order to be consistent with the other metrics. We estimate the DIoSCrete mutual information by binning each dimension of the representations obtained from 10,000 points into 20 bins. Then, the score is computed as follows:

$$MIG = \frac{1}{K} \sum_{k=1}^{K} \left[I(v_{jk}, z_k) - \max I(v_j, z_k) \right]$$

Where zk is a factor of variation, v_j is a dimension of the latent representation, and $jk = \arg\max_j I(v_j, z_k)$.

Time series data often exhibit variations that may not always align with conventional metrics, especially when considering the presence or absence of underlying attributes. To address this challenge, we introduce the Time Disentanglement Score (TDS), a metric designed to assess the disentanglement of attributes in time series data. The foundation of TDS lies in an Information Gain perspective, which measures the reduction in entropy when an attribute is present compared to when it's absent.

$$TDS = \frac{1}{dim(\mathbf{z})} \sum_{n \neq m} \sum_{k} \frac{||z_m - z^{+}_{n,k}||^2}{\mathrm{Var}[z_m]}, \tag{13}$$

In the context of TDS, we augment factor m in a time series window $\mathbf{x}$ with a specific objective: to maintain stable entropy when the factor is present and reduce entropy when it's absent. This augmentation aims to capture the essence of attribute-related information within the data.

B.7 METHOD ABLATION

B.7.1 USING MULTIPLE AUTOENCODERS FOR TRAINING, EACH DEDICATED TO A SPECIFIC LATENT FACTOR.

As discussed in Section 4, we highlighted that employing a multimodal approach does not provide significant advantages in separability tasks. Furthermore, it incurs higher costs compared to Diosco, where all latent variables are funneled into a single unified model. Our hypothesis is grounded in the belief that models equipped with knowledge about the interactions between latent variables can more effectively disentangle them. This advantage outweighs the potential time cost associated with training each model independently see Table. 8.

Method	**RMSE** ↓	**RMIG** ↓	**TDS** ↓	Time ↓
Multi-Diosc (6 model)	1.459	0.917	0.931	42h34min
Multi-Diosc (6 model jointly train)	**0.629**	**0.824**	**0.731**	**29h11min**
Single Diosc	**0.429**	**0.753**	**0.631**	**26h14min**

Table 7: Average Normalized RMSE, RMIG, and TDS Scores for Variants Diosc and case training multi-models for each appliances. (↓ lower values are better [**Top-1**,**Top-2**], the Red row the worst on average, and the Blue the best).

B.7.2 ABLATION ON OTHER PAIRS CORRELATION

Sc.	**Methods**	**No Corr** $\sigma = \infty$			**Pairs: 3** $\sigma = 0.5$			**Pairs: 4** $\sigma = 0.6$			**Random Pair** $\sigma = 0.7$		
	Metrics –>	**DCI** ↓	**TDS** ↓	**RMSE** ↓	**DCI** ↓	**TDS** ↓	**RMSE** ↓	**DCI** ↓	**TDS** ↓	**RMSE** ↓	**DCI** ↓	**TDS** ↓	**RMSE** ↓
	Bert4NILM	-	-	56.4 ±2.58	-	-	70.2±1.45	-	-	72.08±0.96	-	-	70.92±1.15
	S2S	-	-	54.3 ±3.12	-	-	69.5±3.56	-	-	72.31±2.45	-	-	69.95±3.26
	β-VAE	72.4±3.10	0.96±.15	48.6 ±2.32	72.4±3.10	0.96±.15	52.6 ±2.31	72.4±3.10	0.96±.15	54.73±1.54	74.29±2.04	1.08±.09	52.99±1.91
A	β-TCVAE	78.0±1.09	0.94±.13	43.2 ±2.23	78.0±1.09	0.94±.13	49.2 ±1.13	77.23±0.76	0.94±.13	50.87±1.17	79.74±0.84	1.07±.11	49.65±1.43
	FactorVAE	68.4±2.41	0.97±.03	47.7 ±1.35	68.4±2.41	0.97±.03	53.2 ±1.02	69.78±1.43	0.97±.03	54.32±0.64	69.95±1.63	1.00±.02	53.45±0.82
	HFS	79.8 ± .10	0.64 ± .05	57.2 ± 2.15	79.8 ± .10	0.64 ± .05	61.3 ± 1.82	79.56±0.28	0.64±.05	62.33±1.23	80.37±.05	0.72±.03	61.64±1.52
	β-VAE + HFS	73.1±1.01	0.69±.02	34.4±1.89	73.1±1.01	0.69±.02	38.1±1.34	73.59±0.86	0.69±.04	39.65±0.87	74.25±0.59	0.73±.05	38.48±1.04
	β-TCVAE + HFS	**67.2±2.01**	**0.52±.02**	**24.3 ±1.81**	**67.2±2.01**	**0.52±.02**	**27.4 ±1.13**	**67.51±1.84**	**0.52±.07**	**28.94±0.66**	**68.79±1.27**	**0.58±.04**	**27.77±0.83**
	Diosc	**63.5±1.35**	**0.49±.02**	**19.6±1.95**	**69.3±1.2**	**0.4±.02**	**22.3±1.79**	**70.3±0.82**	**0.49±.02**	**23.97±1.19**	**67.12±0.91**	**0.51±.01**	**22.63±1.49**
	Bert4NILM	-	-	57.85 ±1.88	-	-	68.8±1.12	-	-	73.41±1.35	-	-	72.78±0.88
	S2S	-	-	56.38 ±2.22	-	-	67.8±2.76	-	-	73.95±1.91	-	-	70.92±2.25
	β-VAE	73.78±2.68	1.08±.09	50.14 ±1.87	75.47±1.98	0.82±.10	51.7±1.79	70.8±2.62	0.85±.11	55.98±1.27	76.18±1.54	1.16±.08	54.83±1.58
B	β-TCVAE	79.57±0.84	1.07±.11	45.72 ±1.68	80.23±0.54	0.81±.09	48.3±0.94	76.2±0.54	0.83±.10	51.74±0.94	80.88±0.53	1.15±.10	51.15±1.10
	FactorVAE	70.14±1.89	1.00±.02	49.02 ±1.05	71.89±1.24	0.94±.02	52.4±0.85	68.7±1.13	0.92±.02	55.24±0.42	71.57±1.27	1.06±.01	54.68±0.64
	HFS	80.12±.05	0.72±.03	58.49 ±1.45	80.26±.03	0.56±.03	6.0±1.42	78.8±0.15	0.58±.03	63.79±0.97	80.61±.02	0.80±.02	63.22±1.17
	β-VAE + HFS	74.47±0.61	0.73±.05	36.09±1.25	75.12±0.41	0.67±.02	37.4±1.04	72.8±0.52	0.64±.03	40.92±0.66	75.07±0.43	0.75±.03	39.68±0.80
	β-TCVAE + HFS	**68.54±1.36**	**0.58±.04**	**25.88 ±1.20**	**69.28±1.01**	**0.46±.01**	**26.7±0.88**	**66.7±1.51**	**0.45±.02**	**29.82±0.51**	**7.04±0.93**	**0.72±.02**	**40.49±0.64**
	Diosc	**64.42±0.96**	**0.51±.01**	**21.35 ±1.80**	**65.11±0.66**	**0.39±.01**	**21.5±1.44**	**69.5±0.43**	**0.48±.01**	**24.94±0.87**	**65.05±0.71**	**0.55±.01**	**24.05±1.30**
	S2S	-	-	56.28 ±2.43	-	-	73.8±3.91	-	-	74.76±3.75	-	-	73.47±4.12
	β-VAE	74.17±2.01	1.03±.09	50.18 ±1.92	73.84±1.56	0.72±.12	55.7±2.47	76.1±3.36	1.07±.17	56.32±2.31	73.95±1.93	1.16±0.11	55.90±2.40
	β-TCVAE	79.21±0.89	0.98±.10	45.11 ±2.03	79.48±0.75	0.78±.08	50.9±1.27	78.85±0.94	1.05±.15	51.19±1.84	80.57±0.95	1.10±0.11	51.17±1.85
C	FactorVAE	70.23±1.70	0.99±.02	49.12 ±1.18	69.75±1.53	0.99±.03	56.4±1.11	70.92±1.58	0.99±.05	55.48±1.25	70.43±1.74	1.05±.02	54.61±1.34
	HFS	8.04±.06	0.67 ± .03	59.04 ±1.74	80.11±.05	0.60±.04	62.9±1.98	79.91±0.36	0.69±.07	63.52±1.94	80.42±.06	0.73±.03	63.83±2.01
	β-VAE + HFS	74.03±0.79	0.70±.01	35.65±1.59	74.14±0.82	0.74±.01	40.5±1.49	74.26±0.95	0.71±.06	40.32±1.38	74.84±0.51	0.78±.05	39.38±1.19
	β-TCVAE + HFS	**69.04±1.45**	**0.54±.01**	**25.85 ±1.45**	**68.37±1.31**	**0.47±.01**	**28.9±1.28**	**69.07±2.02**	**0.59±.09**	**30.38±1.24**	**69.84±1.43**	**0.62±.04**	**29.29±1.13**
	Diosc	**64.87±1.07**	**0.50±.01**	**19.6±1.95**	**70.54±0.60**	**0.50±.01**	**21.1±1.92**	**71.2±0.94**	**0.44±.03**	**26.97±1.04**	**67.72±1.01**	**0.57±0.01**	**24.12±1.58**

Table 8: Average scores DCI, TDS, and RMSE vary from No Correlation (left) to every appliance correlated with one confounder (right) on uncorrelated test data. Red to blue, with bold indicating the best performance per correlation. (↓ lower is better, ↑ higher is better [**Top-1**,**Top-2**]).

B.8 PSEUDOCODE - DIOSC COSINE SIMILARITY

```
for X_data in loader : #Mini-batch
    X, X_aug = augmented(X_data) # Find appliance activated
    Z = f_phi(X)
    Z_aug = f_phi(X)
    loss = torch.tensor(.0, requires_grad=True)
    for j in range(i+1, M):
        # Select the dimensions i and j from Z
        Z_i_j = torch.stack((Z[:, i], Z[:, j]), dim=-1)
        Z_aug_i_j = torch.stack((Z_aug[:, i], Z_aug[:, j]), dim=-1)
        # Compute the Cartesian product of the selected dimensions
        # Iterate over pairs in the Cartesian product
        for k in list(product(*Z_i_j.T):
            support = torch.cat((k[0], k[1]), dim=0)
            # Iterate over data points in Z_aug_i_j
            for m in Z_aug_i_j.T:
                cos_sim = torch.nn.functional.cosine_similarity(support,
                                                                m, dim=-1)
                loss += cos_sim
    loss.backward()
    optimizer.step()
```

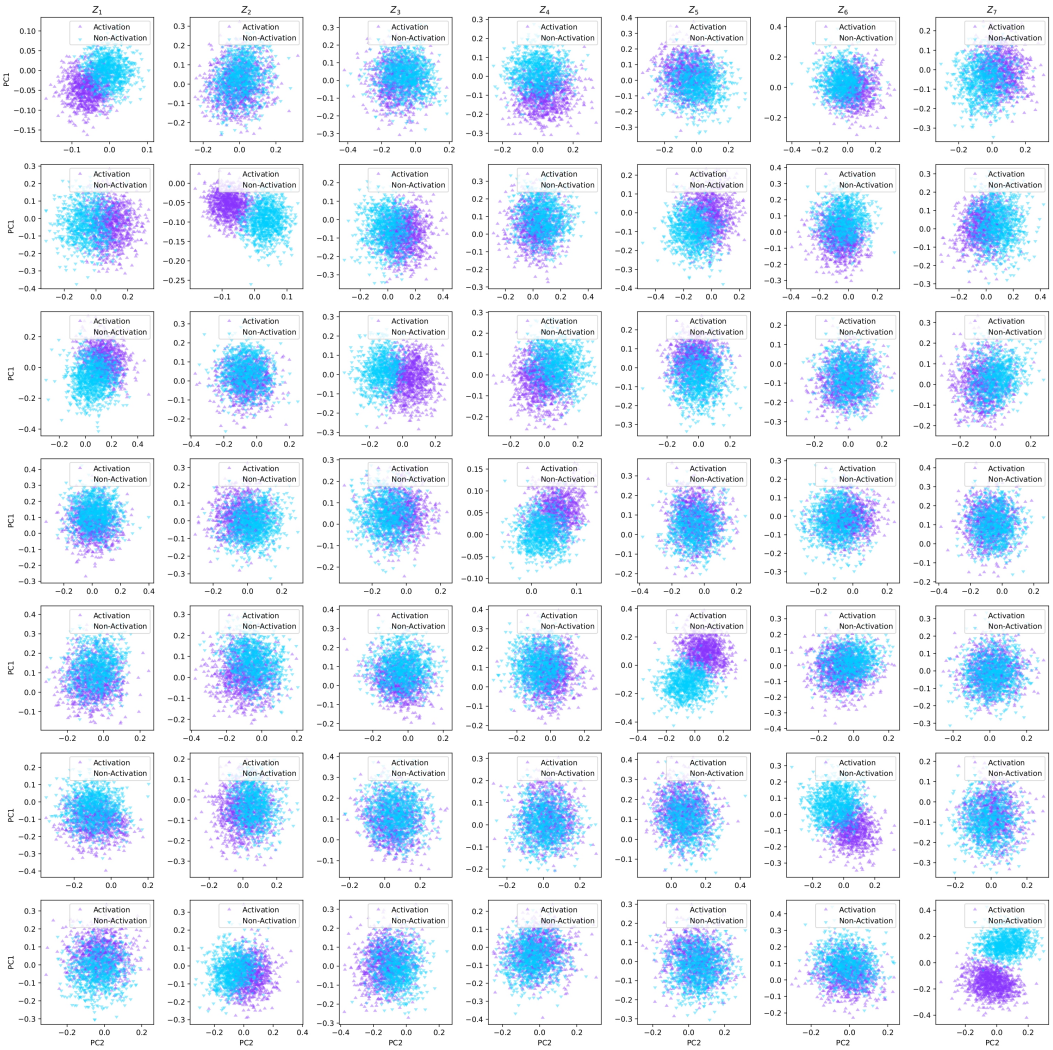

Figure 13: PCA plot of learned representations on Scenario A for appliance. Labeled per activation and non activation.

C CONNECTION BETWEEN DIOSC, CAUSALITY-REPRESENTATION AND THE INFORMATION BOTTLENECK PRINCIPLE

C.1 DISENTANGLEMENT BASED INDEPENDENCE-OF-SUPPORT VIA CONTRASTIVE

The information bottleneck principle applied to disentanglement posits that the objective of disentangling is to learn a representation $\mathbf{z}$ which is informative about the sample but invariant (i.e., uninformative) to the specific distortions that are applied to this sample.

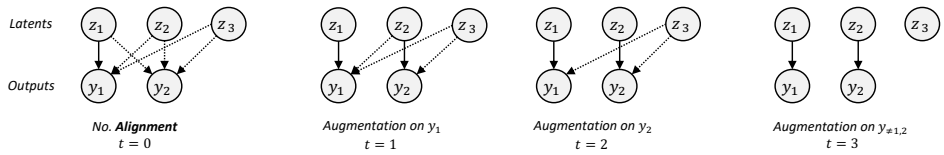

Figure 14: Graphical Model Model Alignment steps of Diosc for case $M = 2$, and $K = 1$.

Disentangling based Independence-of-Support via contrastive can be viewed as a specific instantiation of the information bottleneck objective. We explore in this appendix the connection between DIoSC' loss function and the Information Bottleneck (IB) principle (Ash, 2012). As a reminder, the Diosc loss function is given by:

$$\mathcal{L}_{\text{Diosc}} = \underbrace{\eta \sum_{m} \sum_{\text{negative pairs}} \mathcal{D}(z_m, z_m^-)^2}_{(i)} + \underbrace{\sum_{m} \sum_{\text{positve pairs}} \left(1 - \mathcal{D}(z_m, z_m^+)\right)^2}_{(ii)}, \tag{14}$$

Applied to disentanglement, the IB principle posits that a desirable representation should be as informative as possible about the sample represented while being as invariant (i.e., non-informative) as possible to distortions of that sample, the data augmentations used. This trade-off is captured by the following loss function:

$$\mathcal{IB}_\theta = I(\mathbf{z}, \mathbf{y}) - \beta I(\mathbf{z}, \mathbf{x}) \tag{15}$$

where $I(.,.)$ denotes mutual information, and β is a positive scalar trading off the desideratas of preserving information and being invariant to distortions.

Using a classical identity for mutual information, we can rewrite equation as:

$$\mathcal{IB}_\theta = [H(\mathbf{z}) - H(\mathbf{z}|\mathbf{y})] - \beta[H(\mathbf{z}) - H(\mathbf{z}|\mathbf{x})] \tag{16}$$

where $H(.)$ denotes entropy. The conditional entropy $H(\mathbf{z}|\mathbf{y})$—the entropy of the representation conditioned on a specific distorted sample cancels to 0 because the function g_θ is deterministic, and so the representation $\mathbf{z}$ conditioned on the input sample $\mathbf{y}$ is perfectly known and has zero entropy. Since the overall scaling factor of the loss function is not important, we can rearrange equation Eq. 16 as:

$$\mathcal{IB}_\theta = \underbrace{\frac{1-\beta}{\beta} H(\mathbf{z})}_{(i)} + \underbrace{H(\mathbf{z}|\mathbf{x})}_{(ii)}, \tag{17}$$

Measuring the entropy of a high-dimensional signal generally requires vast amounts of data, much larger than the size of a single batch. In order to circumvent this difficulty, we make the simplifying assumption that the representation $\mathbf{z}$ is distributed as a Gaussian. The entropy of a Gaussian distribution is simply given by the logarithm of the determinant of its covariance function(Ash, 2012). The loss function becomes:

$$\mathcal{IB}_\theta = \mathbb{E}_{\mathbf{x}} \log |\mathcal{D}_{\mathbf{z}|\mathbf{x}}| + \frac{1-\beta}{\beta} \log |\mathcal{D}_{\mathbf{z}}| \tag{18}$$

This equation is still not exactly the one we optimize for in practice. Indeed, our loss function is only connected to the IB loss given by Eq. 18 through the following simplifications and approximations:

- For $\beta < 1$, the optimal solution to the IB trade-off, as indicated by Eq. 18, sets the representation as a constant independent of the input. This scenario results in uninteresting representations and can be disregarded.
- For $\beta \geq 1$, by replacing $\frac{1-\beta}{\beta}$ with a new positive constant η preceded by a negative sign, the second term in Eq. 18 can be simplified. Augmentation of factor while maintaining a fixed other enables the capture of information through this term and simultaneously eliminates non-useful redundancies in the information by the first term.
- In practical terms, direct optimization of the determinant of covariance matrices proves ineffective. Instead, we replace the second term in the loss of Eq. 18 by proxy involves minimizing the Frobenius norm of the cosine similarity, this minimization influences only the off-diagonal terms of the covariance matrix.