Towards Learning Disentangled Representations for Time Series

Yuening Li
Texas A&M University
NEC Labs America

Zhengzhang Chen
NEC Labs America

Daochen Zha
Rice University

Mengnan Du
Texas A&M University

Jingchao Ni
NEC Labs America

Denghui Zhang
NEC Labs America

Haifeng Chen
NEC Labs America

Xia Hu
Rice University

ABSTRACT

Promising progress has been made toward learning efficient time series representations in recent years, but the learned representations often lack interpretability and do not encode semantic meanings by the complex interactions of many latent factors. Learning representations that disentangle these latent factors can bring semantic-rich representations of time series and further enhance interpretability. However, directly adopting the sequential models, such as Long Short-Term Memory Variational AutoEncoder (LSTM-VAE), would encounter a Kullback–Leibler (KL) vanishing problem: the LSTM decoder often generates sequential data without efficiently using latent representations, and the latent spaces sometimes could even be independent of the observation space. And traditional disentanglement methods may intensify the trend of KL vanishing along with the disentanglement process, because they tend to penalize the mutual information between the latent space and the observations. In this paper, we propose **D**isentangle **T**ime-**S**eries (DTS), a novel disentanglement enhancement framework for time series data. Our framework achieves multi-level disentanglement by covering both individual latent factors and group semantic segments. We propose augmenting the original VAE objective by decomposing the evidence lower-bound and extracting evidence linking factorial representations to disentanglement. Additionally, we introduce a mutual information maximization term between the observation space to the latent space to alleviate the KL vanishing problem while preserving the disentanglement property. Experimental results on five real-world IoT datasets demonstrate that the representations learned by DTS achieve superior performance in various tasks with better interpretability.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Machine learning approaches**.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539140

KEYWORDS

time series analysis, deep generative model, interpretable representation, domain adaptation, disentangled representation learning

ACM Reference Format:
Yuening Li, Zhengzhang Chen, Daochen Zha, Mengnan Du, Jingchao Ni, Denghui Zhang, Haifeng Chen, and Xia Hu. 2022. Towards Learning Disentangled Representations for Time Series. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539140

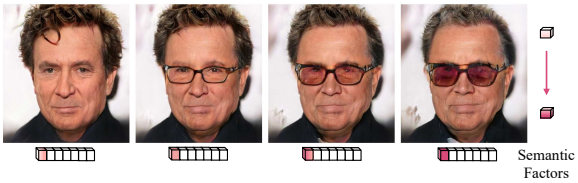
1 INTRODUCTION

Unsupervised representation learning, as a fundamental task of time series analysis, aims to extract low-dimensional representations from complex raw time series without human supervision. Recently, deep generative models have shown great representation ability in modeling complex underlying distributions of time series data. The most representative ones include Long Short-Term Memory Variational AutoEncoder (LSTM-VAE) and its variants [19–21, 27].

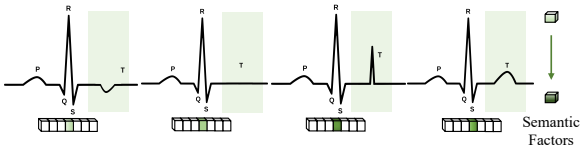
While these representation learning techniques can achieve good performance in many downstream applications, the learned representations often lack the interpretability to expose tangible semantic meanings. In many cases, especially in high-stakes domains, an interpretable representation is critical for diagnosis or decision-making. For example, learning interpretable and semantic-rich representations can help decompose the electrocardiogram (ECG) into cardiac cycles with recognizable phases as independent factors. Furthermore, extracting and analyzing common sequential patterns (*i.e.*, normal sinus rhythms) from massive ECG records can assist clinicians with better understanding irregular symptoms. In contrast, diagnostic processes without transparency or accurate explanations may lead to suboptimal or even risky treatments.

To extract semantically meaningful representations, researchers in computer vision have turned to disentanglement learning, which decomposes the representations into subspaces and encodes them as separate dimensions [13]. A disentangled representation can be defined as one where latent units are sensitive to changes in a single latent factor while being relatively invariant to changes in other factors. Different dimensions in the latent space are probabilistically independent. Fig. 1 (a) shows a semantic factor controls the eyeglasses in the image. Learning factors of variations in the images reveals semantic meanings in the underlying distribution [29].

Motivated by the success of disentanglement in the image domain, in this work, we explore disentangled representations for



(a) Semantic factors for images, where a semantic factor (pink) controls the eye glasses of a facial image [29].



(b) Semantic factors of time series, where a semantic factor (green) controls the sequential trend. Inverted (first), flattened (second), or sharp (third) T waves could provide insights into the abnormalities of the ventricular re-polarization and de-polarization comparing with the normal case (last) [25].

Figure 1: Two traversal plot examples of disentanglement.

time series. Fig. 1 (b) shows an example of how the learned semantic factor can control the shape of ECG time series. Medically, inverted, biphasic, or flattened T wave, as one sequential pattern, could provide insights into the abnormalities of the ventricular repolarization ventricular depolarisation (in green) [25]. In addition, the QT interval, as a group of individual patterns from the beginning of the Q wave to the end of the T wave, could represent the physiologic reactions for the ventricles of the heart to de-polarize and re-polarize. Thus, there exists a vital need for methods that can enhance the interpretability of time series representations from the perspectives of both single factor and group-level factor disentanglement.

However, disentangled representation learning on time series presents several unique challenges. Firstly, *temporal correlations makes the latent representations hard to interpret.* Time series data usually contain temporal correlations, which cannot be directly captured and interpreted by traditional image-focused disentanglement methods [8, 28, 29]. While traditional sequential models, like LSTM or LSTM-VAE [11, 27], could be used to model the temporal correlations, they neither provide interpretable predictions, as is often criticized, nor have a disentanglement mechanism. Secondly, *naively applying disentanglement methods to sequential models may intensify the Kullback–Leibler (KL) vanishing problem.* When compounded with strong autoregressive decoders, VAE based sequential models often converge to a degenerated local optimum known as "KL vanishing", which causes the latent variables to be relatively independent of the observations [32]. Unfortunately, traditional disentanglement methods may intensify the trend of KL vanishing along with the disentanglement process, because they tend to penalize the mutual information between the latent space and the observations (detailed analysis of this problem is provided in Section 2.2 and Section 2.3). Thirdly, *interpretable semantic concepts often rely on multiple factors instead of individuals.* A human-understandable sequential pattern, called a semantic component, is usually correlated with multiple factors. It is hard to interpret time series with a single latent factor.

To address these challenges, we propose **D**isentangle **T**ime-**S**eries (DTS) for learning semantically interpretable time series representations. To the best of our knowledge, DTS is the first attempt to incorporate disentanglement strategies for time series. In particular, we design a multi-level disentanglement strategy that accounts for both individual factors and group-level segments, to generate hierarchical semantic concepts as the interpretable and disentangled representations of time series. To disentangle individual latent factors, DTS adjusts the training objective from two aspects: 1) augmenting the original training objective by decomposing the evidence lower bound, which aims to preserve the disentanglement property and alleviate the KL vanishing problem simultaneously; 2) introducing a mutual information maximization term, which aims to preserve the correlation between the latent variables and the original time series. In addition, we theoretically prove that the new objective can balance the preference between correct inference and fitting data distribution. To disentangle group-level semantic segments, DTS learns to decompose time series into independent semantic segments via applying gradient reversal layers on irrelevant tasks. Each of the semantic segments contains batches of independent latent variables. The segments with target task-relevant information are utilized to eliminate negative transfer from incidentally encoded irrelevant information.

The contributions of this work are summarized as follows:

- We introduce a novel and challenging real-world problem (*i.e.*, disentangling time series) and propose DTS to incorporate disentanglement strategies for time series representation learning.
- We propose a multi-level time series disentanglement strategy, covering both individual latent factor and group-level semantic segments, to generate hierarchical semantic concepts as the interpretable and disentangled representation of time series.
- We introduce an evidence lower bound decomposition strategy that could balance the preference between correct inference and data distribution fitting. We theoretically show how to preserve the correlation between the latent space and inputs and factorize the latent space for disentanglement simultaneously.
- The proposed DTS framework is extensively evaluated on on five real-world IoT datasets. The results show that DTS provides more meaningful disentangled representations of time-series, and is quantitatively effective for downstream tasks.

2 METHODOLOGY

In this section, we propose DTS, a multi-level disentanglement approach (see Fig. 2) to enhance time series representation learning. The key idea of DTS is to factorize the latent space as independent semantic concepts. DTS mainly consists of *Individual Factor Disentanglement*, and *Group Segment Disentanglement*. The Individual Factor Disentanglement module decomposes the latent variables into independent factors that contain different semantic meanings, while the Group Segment Disentanglement module aims to enrich the group-level semantic meaning of sequential data by grouping them into a batch of segments. To achieve the multi-level disentanglement, a novel evidence lower bound (ELBO) decomposition strategy is proposed to find evidence linking factorial representations to disentanglement without sacrificing the correct inference.

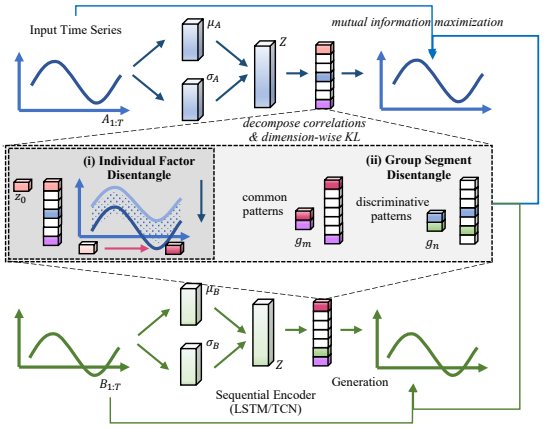


Figure 2: DTS consists of two components: (i) individual factor disentanglement: to learn semantic factors like z_0 (shown as pink) to control the sequential pattern of the time series, *e.g.*, the time series moves down when adjusting z_0; and (ii) group segment disentanglement: to learn more complex semantic patterns g_m, g_n (illustrated as the common and discriminative patterns). A, B denote two time series.

Notations: Let $\mathbf{x} = [x_1, x_2, ..., x_T] \in \mathbb{R}^T$ be a time series of length T, which is associated with a latent representation $\mathbf{z} = [z_1, z_2, ..., z_n] \in \mathbb{R}^n$. Each entry z_i is a value of a latent variable, which is a disentangled factor that describes a particular sequential pattern of $\mathbf{x}$. The set $Z_s = \{z_1, z_2, ..., z_n\}$ contains all of the factors. As some complex patterns may only be described by a sub-group of factors from Z_s, we use $Z_g = \{g_1, ..., g_m\}$ to denote a division of Z_s, where g_i includes several latent variables from Z_s, *i.e.*, $g_i \subset Z_s$, and the sub-groups are disjoint, *i.e.*, $g_i \cap g_j = \emptyset, \forall 1 \leq i, j \leq m$, and $m \leq n$.

Specifically, a disentangled factor z_i should be sensitive to the changes in a single semantic concept that governs the generation of the time series, while being invariant to the changes caused by other latent variables in Z_s [6]. For example, in Fig. 1(b), one latent variable controls the shape of the time series in the green interval but will not cause the changes of other intervals (which could be controlled by other latent variables). We denote such disentanglement between factors by $z_i \perp\!\!\!\perp z_j$. Similarly, two groups of factors are disentangled, *i.e.*, $g_i \perp\!\!\!\perp g_j$, if they are invariant to the changes of the other's corresponding sequential patterns.

Problem: Given a training dataset $\mathcal{D} = \{\mathbf{x}\}$, our goal is to solve a *multi-level* time series disentanglement problem, by learning: (1) a set of latent variables $Z_s = \{z_1, z_2, ..., z_n\}$, where $z_i \perp\!\!\!\perp z_j, \forall 1 \leq i, j \leq n$; and (2) a division of latent variables $Z_g = \{g_1, ..., g_m\}$, where $g_i \perp\!\!\!\perp g_j, \forall 1 \leq i, j \leq m$, such that the latent representation $\mathbf{z}$ of each time series $\mathbf{x}$ is semantically meaningful.

2.1 Preliminaries

First, we introduce how disentanglement is achieved for static data from a generative modeling perspective [6]. A latent variable generative model defines a joint distribution between a feature space $Z \in \mathcal{Z}$, and the observation space $\mathbf{x} \in \mathcal{X}$. Suppose $p(Z)$ is a prior distribution of the latent variables, and $p_\theta(\mathbf{x} \mid Z)$ is a conditional probability of $\mathbf{x}$ that is parameterized by neural networks θ (*e.g.*, RNNs), then the disentanglement goal is to maximize the marginal likelihood of the observed samples in the training dataset:

$$\mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] = \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}\left[\log \mathbb{E}_{p(Z)}\left[p_\theta(\mathbf{x} \mid Z)\right]\right]. \quad (1)$$

where $p_\mathcal{D}(\mathbf{x})$ represents the true underlying distribution, which can be estimated using the training dataset.

However, exact posterior inference of Eq. (1) is analytically intractable, due to the integration $\mathbb{E}_{p(Z)}\left[p_\theta(\mathbf{x} \mid Z)\right] = \int_z p_\theta(\mathbf{x} \mid Z)p(Z)\mathrm{d}z$ over latent variables. Therefore, similar to variational inference [17], an amortized inference distribution $q_\phi(Z \mid \mathbf{x})$ is introduced to approximate the posterior with learnable parameters ϕ. A lower bound (ELBO) of Eq. (1) can be derived as:

$$\mathcal{L}_{\text{ELBO}}(\mathbf{x}) = -D_{\text{KL}}\left(q_\phi(Z|\mathbf{x}) \| p(Z)\right) + \mathbb{E}_{q_\phi(Z|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|Z)\right]. \quad (2)$$

To learn disentangled representations, β-VAE [6, 14] has been introduced as an effective solution. It is a variant of the Variational AutoEncoder (VAE) that attempts to learn a disentangled representation by optimizing a heavily penalized objective with $\beta > 1$.

$$\mathcal{L}_{\beta-\text{ELBO}}(\mathbf{x}) = -\beta D_{\text{KL}}\left(q_\phi(Z|\mathbf{x}) \| p(Z)\right) + \mathbb{E}_{q_\phi(Z|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|Z)\right]. \quad (3)$$

The penalization enables disentangled effects of models on image datasets. The β constraint imposes a limit on the capacity of the latent information channel and controls the emphasis on learning statistically independent latent factors. With increasing β, the latent variables become more disentangled as the distributions in the latent space deviate from each other by fitting the marginal Gaussian distribution more than the KL divergence. Thus, semantically similar observations move closely, resulting in clusters corresponding to underlying factors of variation, which facilitate interpretation.

2.2 Sequential Data Meets VAE: KL Vanishing

To model sequential data, the autoregressive decoder is often used with VAE, such as LSTM-VAE [10], for time series analysis. However, when compounded with strong autoregressive decoders such as LSTMs [16], VAE suffers from a critical problem known as posterior collapse or KL vanishing. The decoder in VAE reconstruct the data independently of the latent variables, and the KL term vanishes to 0. This is because the reconstruction term in the objective will dominate the KL divergence term during the training phase. As a result, the model generates time series without making effective use of the latent variables.

Specifically, in Eq. 3, the latent variables Z become independent from observations $\mathbf{x}$, when the KL divergence term collapses to zero. Thus, the latent variable Z can not serve as an effective representation for the input $\mathbf{x}$, which is also known as the *information preference* problem [9]. In this case, pushing Gaussian clouds away from each dimension in the latent space to encourage disentangling latent factors becomes meaningless if latent distributions are independent and unhooked with the observation space.

2.3 Individual Factor Disentanglement

To alleviate the KL vanishing problem and preserve the disentanglement property, in this section, we first decompose the evidence lower bound (ELBO) to better understand the disentaglement and the causes of the KL vanishing problem. Then, we introduce the mutual information maximization term to the ELBO decomposition, which enables learning a better representation Z that captures the semantic characteristics of the input $\mathbf{x}$.

2.3.1 ELBO TC-Decomposition. To understand the internal mechanism of the disentanglement, we decompose the ELBO to find evidence linking factorial representations to disentanglement. By decomposing the ELBO into separate components, we can have a new perspective for the cause of the KL vanishing problem: *introducing heavier penalty on the ELBO tends to encourage the independence between latent variables, but neglects the mutual information between the latent variables and the input.*

More specifically, we define $q_\phi(Z, \mathbf{x}) = q_\phi(Z \mid \mathbf{x})p_\theta(\mathbf{x})$. Following [8, 24], we denote $q_\phi(Z) = \mathbb{E}_{p_\theta(\mathbf{x})} q(z \mid \mathbf{x})$ as the aggregated posterior, which captures the aggregate structure of the latent variables under the data distribution of $p_\theta(\mathbf{x})$. Mathematically, the KL term in Eq. 2 and 3 can be decomposed with a factorized $p(Z)$ as:

$$D_{\mathrm{KL}}(q_\phi(Z \mid \mathbf{x})||p(Z)) = \underbrace{\mathrm{KL}(q_\phi(Z, \mathbf{x})||q_\phi(Z)p_\theta(\mathbf{x}))}_{\text{(i) Index-Code MI}} + \underbrace{\mathrm{KL}(q_\phi(Z)\|\prod_j q_\phi(z_j))}_{\text{(ii) Total Correlation}} + \underbrace{\sum_j \mathrm{KL}\left(q_\phi(z_j)\|p(z_j)\right)}_{\text{(iii) Dimension-wise KL}} \quad (4)$$

where z_j denotes the jth dimension of the latent variable.

The first term can be interpreted as the index-code mutual information (MI) $I_{q_\phi}(Z; \mathbf{x})$, which is the MI between the data variable and latent variable. The second term is referred to as the total correlation (TC), which acts as a generalization of MI to more than two random variables [34]. TC also evaluates the dependency between the variables. The penalty on TC encourages statistically independent factors in the data distribution. A heavier penalty on this term induces a more disentangled representation. This term explains the success of β -VAE. Recent works [8, 36] indicate TC is the most important term in this decomposition for learning disentangled representations by only penalizing on this term. The last term is the dimension-wise KL, which prevents individual latent dimensions from deviating too far away from their priors. It serves as a complexity penalty on the aggregate posterior, according to the minimum description length formulation of the ELBO [15].

Increasing the β may intensify the KL vanishing problem: along with optimizing the ELBO, when the model has a better quality of disentanglement within the learned latent representations, it penalizes the MI simultaneously. It can, in turn, lead to under-fitting or ignoring the latent variables. The approximate inference distribution is often significantly different from the true posterior. This is undesirable because one major goal of unsupervised learning is to learn meaningful latent features that should depend on the observations. Thus, the ELBO objective favors fitting the data distribution over performing correct amortized inference. When the two goals are conflicting, the ELBO objective tends to sacrifice the correct inference to better fit (or worse overfit) training data, which we can refer to as the *information preference* problem.

2.3.2 ELBO DTS-Decomposition. To address the information preference problem, in this subsection, we propose an ELBO decomposition strategy by explicitly maximizing the MI between the latent space and the input. In this way, we can disentangle the latent space without sacrificing the correct inference.

Specifically, as discussed before, the latent variable Z becomes independent from observations $\mathbf{x}$. To encourage the model to use the latent variables, we add an MI maximization term, which encourages a high MI between $\mathbf{x}$ and Z. In other words, we can address the information preference problem by balancing the preference between correct inference and fitting data. Beginning from the ELBO in LSTM-VAE (in Eq. 2), we arrive at:

$$-D_{\mathrm{KL}}\left(q_\phi(Z \mid \mathbf{x})\|p(Z)\right) + \alpha I_{q_\phi}(\mathbf{x}; Z) + \mathbb{E}_{q_\phi(Z|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid Z)\right] \quad (5)$$

where $I_{q_\phi}(\mathbf{x}; Z)$ denotes the MI between $\mathbf{x}$ and Z under the distribution $q_\phi(\mathbf{x}; Z)$.

But this objective can not be directly optimized. Thus, we rewrite it into another equivalent form:

$$-D_{\mathrm{KL}}\left(q_\phi(Z \mid \mathbf{x})\|p(Z)\right) + \alpha D_{\mathrm{KL}}\left(q_\phi(Z)\|p(Z)\right) + \mathbb{E}_{q_\phi(Z|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid Z)\right]. \quad (6)$$

The MI maximization term (the second part of Eq. 6) plays the same role as the first term in the ELBO-TC decomposition (as shown in Eq. 4), but the optimization directions are contrary. Thus, *increasing the disentanglement degree may intensify the KL vanishing problem, and vice versa.* To enforce the model to preserve the disentanglement property while alleviating the KL vanishing, here, we combine the MI regularizer term with the ELBO-TC decomposition in Eq. 4 and merge the MI maximization term, then the ELBO can be re-written as:

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}}(\mathbf{x}) = &-\beta D_{\mathrm{KL}}(q(Z)\|\prod_j q\left(z_j\right)) - \beta \sum_j D_{\mathrm{KL}}\left(q\left(z_j\right)\|p\left(z_j\right)\right) \\ &+ (\alpha - \beta) D_{\mathrm{KL}}\left(q_\phi(Z)\|p(Z)\right) + \mathbb{E}_{q_\phi(Z|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid Z)\right],\end{aligned} \quad (7)$$

Mathematically, we alleviate the KL vanishing problem by introducing the MI maximization term, while preserving a heavier penalty (when $\beta > 1$) on the total correlation and the dimension-wise KL to keep the disentanglement property.

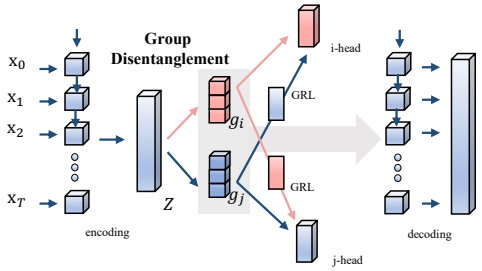


Figure 3: Structure of the group segment disentanglement.

2.4 Group Segment Disentanglement

By employing the aforementioned ELBO DTS-Decomposition, we can achieve individual factor disentanglement. However, the capacity of one single factor is often not sufficient to represent complex concepts [14]. Thus, in this subsection, we generalize individual disentanglement to group segment disentanglement to further enrich the latent factor representations.

Fig. 3 illustrates the process of learning latent group segment disentanglement. For simplicity, here, we show how to learn two semantic segments, although our method can be extended to more segments. Formally, let g_i and g_j be two semantic segments in Z, where our goal is to make them independent with each other, *i.e.*, $g_i \perp\!\!\!\perp g_j$. To achieve this, we optimize each segment with two objectives to encourage the representations to be semantically independent.

First, we derive an ELBO objective for group segments. Following the evidence lower bound of the marginal likelihood in Eq. 6, we

can get a similar form for group segments:

$$\mathcal{L}_{\text{ELBO-G}}(\mathbf{x}) = -D_{\text{KL}}\left(q_{\phi_m}(g_i \mid \mathbf{x}) \| p(g_i)\right) - D_{\text{KL}}\left(q_{\phi_n}(g_j \mid \mathbf{x}) \| p(g_j)\right) + \mathbb{E}_{q_{\phi_m}(g_i, g_j|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid g_i, g_j)\right] + \alpha D_{\text{KL}}\left(q_\phi(Z) \| p(Z)\right). \tag{8}$$

which (1) approximates the $p(g_i)$ and $p(g_j)$ from $q_\phi(g_i \mid \mathbf{x})$ and $q_\phi(g_j \mid \mathbf{x})$, respectively, (2) fits the data distribution via reconstruction, and (3) maximizes MI between the latent and the input spaces.

Second, we introduce auxiliary classification heads to encourage each segment to contain only a single concept by leveraging the labeling function (*i.e.*, the mapping to the ground truths) of each auxiliary task. Formally, let $f_i : \mathcal{Z} \to \mathcal{Y}$ and $f_j : \mathcal{Z} \to \mathcal{Y}$ be the labeling functions of two auxiliary tasks that correspond to g_i and g_j, respectively. That is, $f_i(Z_g)$ and $f_j(Z_g)$ are the ground truths of the two tasks for g_i and g_j. The two classification heads aim to learn hypotheses $h_i : \mathcal{Z} \to \mathcal{Y}$ and $h_j : \mathcal{Z} \to \mathcal{Y}$ to approximate f_i and f_j, respectively. To optimize h_i and h_j, we can quantify the empirical error based on the following theorem.

Theorem 1. *For two independent group segments g_i and g_j, where $g_i \perp\!\!\!\perp g_j$ and $Z_g = \{g_i, g_j\}$, the empirical error on the disentangled segments according to the distribution $\mathcal{Z}$ that a hypothesis h disagrees with a labeling function f is:*

$$\epsilon(h) = \mathrm{E}_{g_i \sim \mathcal{Z}}\left[f_i(Z_g) - h_i(g_i)\right] + \mathrm{E}_{g_j \sim \mathcal{Z}}\left[f_j(Z_g) - h_j(g_j)\right]$$

where $\epsilon(h)$ denotes the empirical error of DTS *with respect to h.*

Proof. Since $g_i \perp\!\!\!\perp g_j$, we can derive the empirical error as follows:

$$\begin{aligned}\epsilon(h) &= \mathrm{E}_{(g_i, g_j) \sim \mathcal{Z}}\left[f(Z_g) - h(Z_g)\right] \\ &= \mathrm{E}_{g_i \sim \mathcal{Z}}\left[f_i(Z_g) - h_i(g_i)\right] + \mathrm{E}_{g_j \sim \mathcal{Z}}\left[f_j(Z_g) - h_j(g_j)\right].\end{aligned}$$

Remark 1. *Based on the independence property between g_i and g_j, the distribution of $\mathcal{Z}$ can be decomposed into two parts so as to the error.*

Following the above objectives, we can learn g_i and g_j as follows. Let θ_i and θ_j be the parameters of the auxiliary classification heads for g_i and g_j, and θ_{vae} be the parameters of the VAE model. Assuming that $P(g_i), P(g_j) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ (which is a common assumption in generative models), we can apply a reparameterization trick by using sequential models (LSTMs or TCNs [3]) as the universal approximator of q to encode the $\mathbf{x}$ into g_i and g_j, respectively. Then, the ELBO objective in Eq. 8 will be applied to learn disentangled group segments. Meanwhile, we can resort to auxiliary classification heads to make g_i task-j-invariant, and g_j task-i invariant:

$$\begin{aligned}\mathbb{E}_{\text{i}}(\theta_{vae}, \theta_i, \theta_j) &= \mathbb{E}\left[h_i(g_i; \theta_{vae}, \theta_i) - f_i(Z_g)\right] - \lambda \mathbb{E}\left[h_j(g_i; \theta_{vae}, \theta_j) - f_j(Z_g)\right] \\ \mathbb{E}_{\text{j}}(\theta_{vae}, \theta_i, \theta_j) &= \mathbb{E}\left[h_j(g_j; \theta_{vae}, \theta_j) - f_j(Z_g)\right] - \lambda \mathbb{E}\left[h_i(g_j; \theta_{vae}, \theta_i) - f_i(Z_g)\right].\end{aligned} \tag{9}$$

Specifically, we optimize the parameters $\hat{\theta_{vae}}, \hat{\theta}_i, \hat{\theta}_j$ based on: $(\hat{\theta_{vae}}, \hat{\theta}_i) = \arg\min_{\theta_{vae}, \theta_i} E\left(\theta_i, \hat{\theta}_j\right)$ and $\hat{\theta}_j = \arg\max_{\theta_j} E\left(\hat{\theta_{vae}}, \hat{\theta}_i, \theta_j\right)$, where the parameter λ controls the trade-off between the two objectives that shape the features during training. The update process is similar to vanilla stochastic gradient descent updates for feed-forward deep models. $-\lambda$ factor can make disentangled features less discriminative for the irrelevant task. Here, we use a gradient reversal layer (GRL) [12] to exclude the discriminative information. During the forward propagation, GRL acts as an identity transform. During the backpropagation, GRL takes the gradient from the subsequent level, and multiplies the gradient by a negative constant, then passes it to the preceding layer.

2.4.1 Application to Domain Adaptation. To further illustrate the benefits of the proposed group segments disentanglement for time series, we apply it to solve the domain adaptation problem as a concrete application scenario. When labeled data is scarce for a specific target task, domain adaptation often offers an effective solution by utilizing data from a related source task from a transfer learning perspective. The hope is that this source domain is related to the target domain, and thus transferring knowledge from the source can improve the performance within the target domain [33]. But "unrelated" features in the source samples can hurt the performance, leading to negative transfer. In this subsection, we take a step towards addressing the negative transfer issue via disentangling the latent variables into grouped "class-dependent" segments that are domain invariant as transferable common knowledge and "domain-dependent" segments that may lead to negative transfer.

In the unsupervised domain adaptation problem, we use the labeled samples $D_S = \left\{\mathbf{x}_i^S, y_i^S\right\}_{i=1}^{n_S}$ on the source domain to classify the unlabeled samples $D_T = \left\{\mathbf{x}_j^T\right\}_{j=1}^{n_T}$ on the target domain. We aim to obtain two independent latent variables with disentanglement, including a domain-dependent latent variable g_d and a class-dependent latent variable g_y. These two variables are expected to encode the domain information and the class information, respectively. Then, we can use the class-dependent latent variable for classification since it is domain-invariant. Under the assumption that there exists some hypothesis h that performs well in both domains, we show that this quantity together with the empirical source error $\epsilon_S(h)$ characterize the target error $\epsilon_T(h)$. Deriving from Theorem 1, we have:

Theorem 2. *Assume that the class factor g_y and the domain factor g_d are independent, i.e., $g_y \perp\!\!\!\perp g_d$. Let $Z_g = \{g_y, g_d\}$, and the error on the disentangled source and target domain with a hypothesis h is:*

$$\begin{aligned}\epsilon_S(h) &= \mathrm{E}_{g_y \sim \mathcal{Z}_S}\left[f_y(Z_g) - h_y(g_y)\right] + \mathrm{E}_{g_d \sim \mathcal{Z}_S}\left[f_d(Z_g) - h_d(g_d)\right] \\ \epsilon_T(h) &= \mathrm{E}_{g_y \sim \mathcal{Z}_T}\left[f_y(Z_g) - h_y(g_y)\right] + \mathrm{E}_{g_d \sim \mathcal{Z}_T}\left[f_d(Z_g) - h_d(g_d)\right].\end{aligned}$$

According to the Theorem 2, we can find that the disentangled empirical classification error rate with respect to h in the source domain is lower than before disentanglement ($\epsilon_S^y(h) = \epsilon_S(h) - \epsilon_S^d(h)$, where $\epsilon_S^d(h) \geq 0$). Here, we prove that the disentanglement of the representation space could be helpful and necessary for obtaining a lower classification error rate. The probabilistic bound on the performance $\epsilon_T(h)$ evaluated on the target domain given its performance $\epsilon_S(h)$ on the source domain can be defined as:

$$\varepsilon_{\mathcal{T}}(h) \leq \varepsilon_{\mathcal{S}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda \tag{10}$$

where $d_{\mathcal{H}\Delta\mathcal{H}}$ measures the discrepancy distance between the source and target distribution with respect to hypothesis h; λ does not depend on a particular h, and is small enough to be a negligible term in the bound [4]. Our method provides a smaller discrepancy distance between two domains since it eliminates the discriminative information during the disentanglement. Thus, a tighter upper bound for the $\varepsilon_{\mathcal{T}}(h)$ can be achieved through reducing $d_{\mathcal{H}_\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T})$, which eventually leads to a better approximation of $\varepsilon_{\mathcal{T}}(h)$.

3 RELATED WORK

Relation to LSTM-VAE and β-VAE: Encoders in LSTM-VAE maps a data point from the observation space into a probabilistic output of Gaussian cloud with mean $\mu(\mathbf{x})$ and 'ellipsoid' orientation determined by the diagonal covariance matrix $\text{diag}(\sigma(\mathbf{x}))$. Compared with LSTM-VAE, β-VAE relaxes the stochastic 'compression' via mapping everything to a Gaussian heap by applying the relaxation parameter that might give more preference to the fitting loss and sacrifice the correct inference. Our proposed DTS can be considered as another form of compression by the minimization of $I_{\boldsymbol{\phi}}(\mathbf{x}; Z)$. DTS relaxes the condition to map all conditional distributions to one Gaussian heap.

Relation to Domain Adaptation Methods: Existing domain adaptation methods focus on (i) utilizing maximum mean discrepancy to measure the domain alignment [31]; and (ii) extracting the domain-invariant representation as transferable common knowledge on the feature space [7, 23, 35]. In contrast, the DTS aims to extract the group segments in the latent disentangled semantic representation of the data. DTS also introduces interpretability in the latent space via weak-supervised signals as imposing special constraints on the latent variables.

4 EXPERIMENTS

In this section, we aim to answer the following questions. **Q1**: Compared with non-disentanglement methods, how quantitatively effective is the proposed disentanglement strategy? **Q2**: Does individual latent factor disentanglement benefit the generation process of the time series in a more informative way? **Q3**: Can latent group segment disentanglement strategy separate semantic concepts?

4.1 Experimental Setup

We provide insights on interpreting the latent representations with semantic meanings. First, to validate the effectiveness of the disentanglement strategy, we apply DTS to domain adaptation tasks, and further illustrate the benefits of DTS on separating and extracting different semantic meaningful sequential patterns as transferable common knowledge and domain-dependent information (Section 4.2); Second, we provide latent traversals to validate that DTS tends to discover more informative latent factors and provide more meaningful disentangled representations of time series (Section 4.3); Finally, we visualize the disentangled segments over the representation space, to investigate the discriminative ability of the group disentanglement strategy (Section 4.4).

4.1.1 Datasets. We evaluate DTS on five benchmark datasets for individual latent factor and group segment disentanglements.

- **Human Activity Recognition (HAR)** [2]: contains sequential accelerometer, gyroscope, and estimated body acceleration data.
- **Heterogeneity Human Activity Recognition (HHAR)** [30]: includes accelerometer data from 31 smartphones of different manufacturers and models positioned in various orientations.
- **WISDM Activity Recognition (WISDM AR)** [18]: contains 33 participants' accelerometer data, which are sampled at 20 Hz.
- **uWave** [22]: is a large gesture library with over 4000 samples collected from eight users over an elongated period of time for a gesture vocabulary with eight gesture patterns.

Problem	W/O	R-DANN	VRADA	CoDATS	DTS	Target
HAR 2 → 11	83.3	80.7	64.1	74.5	**84.3**	100.0
HAR 7 → 13	89.9	75.3	78.3	96.5	**98.1**	100.0
HAR 9 → 18	31.1	56.6	59.8	85.8	**89.8**	100.0
HAR 14 →19	62.0	71.3	64.4	98.6	**100.0**	100.0
HAR 18 →23	89.3	78.2	72.9	89.3	**94.9**	100.0
HAR 7 → 24	94.4	84.8	93.9	99.1	**100.0**	100.0
HAR 17 →25	57.3	66.3	52.0	97.6	**100.0**	100.0
HHAR 1 →3	77.8	85.1	81.3	90.8	**93.7**	99.2
HHAR 3 →5	68.8	85.4	82.3	94.3	**95.9**	99.0
HHAR 4 →5	60.4	70.4	71.6	94.2	**94.9**	99.0
HHAR 1 →6	72.1	81.7	74.9	90.8	**92.1**	98.8
HHAR 4 →6	48.0	64.6	62.7	85.3	**92.3**	98.8
HHAR 5 →6	65.1	54.4	60.0	91.7	**92.5**	98.8
HHAR 5 →8	95.3	82.5	87.5	95.8	**97.9**	99.3
WISDM 4 → 15	78.2	69.2	82.7	81.4	**82.9**	100.0
WISDM 2 → 25	81.1	57.8	72.2	90.6	**95.8**	100.0
WISDM 25 → 29	47.1	61.6	81.9	74.6	**82.2**	95.7
WISDM 7 → 30	62.5	41.7	61.9	73.2	**89.2**	100.0
WISDM 21 → 31	57.1	61.0	68.6	92.4	**96.4**	97.1
WISDM 2 → 32	60.1	49.0	66.7	68.6	**70.7**	100.0
WISDM 1 → 7	68.5	44.8	63.0	66.1	**72.7**	96.4
uWave 2 → 5	86.3	33.3	18.5	98.2	**100.0**	100.0
uWave 3 → 5	82.7	63.7	32.4	92.9	**95.6**	100.0
uWave 2 → 6	86.0	34.5	25.3	93.8	**97.8**	100.0
uWave 2 → 7	85.1	53.9	12.2	91.4	**98.9**	100.0
uWave 3 → 7	95.5	64.0	30.4	92.0	**98.9**	100.0
uWave 1 → 8	100.0	78.6	11.0	93.8	**100.0**	100.0
uWave 7 → 8	95.2	49.7	12.5	93.8	**96.7**	100.0

Table 1: Target classification accuracy (based on the class-dependent representation g_y) for time series domain adaptation (shown as [Dataset source_id → target_id]: accuracy) on randomly-chosen problems for each dataset, adapting between different users. We include no adaptation as an approximate lower bound (W/O), and models trained directly on labeled target data as upper bound (target).

- **ECG Signal** [25]: contains heartbeats annotated by at least two cardiologists. The annotations are mapped into 5 groups.

Our adaptation problems consist of the realistic use-case adapting a model from one participant's data to another participant's data. Each dataset consists of data from a number of participants. The multivariate time series datasets include a participant identifier, and we use this feature to split data into multiple domains. We build up a classifier from one source domain with labeled data, and apply it for the label-free target domain as adaptation. We follow the same procedure in [35] to select 7 of the possible adaptation problems between two domains as [source_id→ target_id] (excluding adapting a domain to itself).

4.1.2 Baselines. We compare DTS with three state-of-the-art domain adaptation algorithms (R-DANN, VRADA, and CoDATS) and one time series generative model (LSTM-VAE):

- **Recurrent Domain Adversarial Neural Network(R-DANN)** [1]: employs an LSTM network, and promotes the emergence of features that are (i) discriminative for the learning task on the source domain and (ii) indiscriminate with respect to the shift between the domains.
- **Variational Recurrent Adversarial Deep Domain Adaptation (VRADA)** [26]: uses a variational RNN and trains adversarially to capture temporal relationships that are domain-invariant.
- **Convolutional Deep Domain Adaptation (CoDATS)** [35]: leverages domain-invariant domain adaptation methods to operate on time series data, and utilizes weak supervisions from labels.

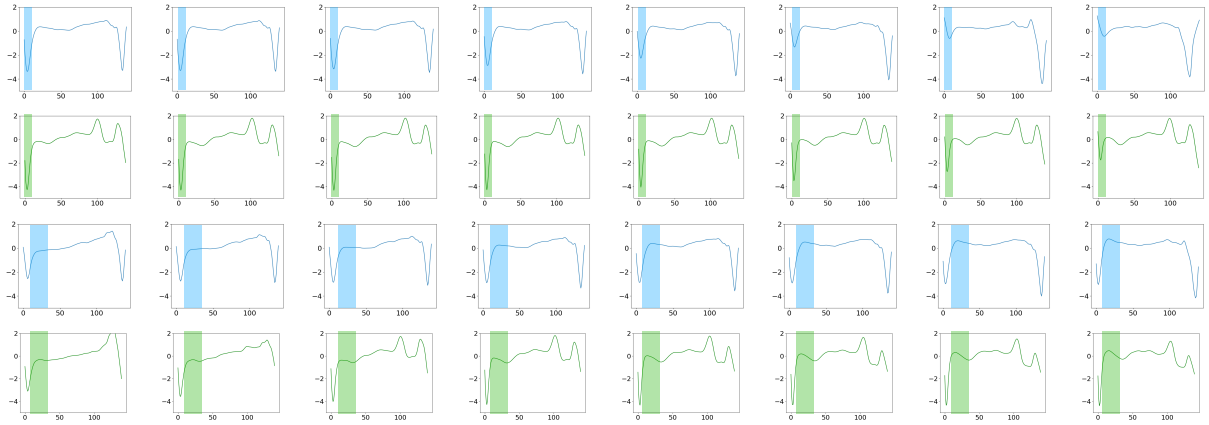


Figure 4: Latent traversal plots from DTS on ECG. All figures of latent variables traversal each block corresponds to the traversal of a single latent variable while keeping others fixed to their inferred. Each row represents a different seed image used to infer the latent values with traversal over the [-4, 4] range. Blue and green denote two time series with different sequential patterns. The first two rows denote the decline degree at the first turning point, transition from rigid to a more mild manner; The last two rows denote the rising trend at the second turning point, transition from inconspicuous to obvious.

Dataset	Acc/D-S	Acc/D-T	Acc/C-S	Acc/C-T	AUC/C-S	AUC/C-T
HAR	100.0	100.0	100.0	97.2	100	99.2
HAR/r	54.7	66.7	35.9	22.2	54.6	53.7
HHAR	96.9	98.8	97.9	91.1	99.8	98.7
HHAR/r	70.8	70.9	30.5	17.6	58.3	52.7
WISDM	100	100	94.5	70.7	99.0	85.9
WISDM/r	67.3	46.7	38.2	33.3	69.7	67.9
uWave	100	100	99.1	94.4	99.0	85.9
uWave/r	55.4	48.2	18.8	12.5	54.6	53.7

Table 2: Ablation studies: discriminability. The results are listed as X-Y (X: D denotes discriminability on domains, C denotes discriminability on tasks; Y: S and T denote features sampled from source and target domains, respectively). Top rows: performance based on the disentangled domain-dependent g_d, or class-dependent g_y for time series domain adaptation. Bottom rows(/r): performance based on the domain-invariant g_y and class-invariant g_d segments.

- **LSTM-VAE** [10]: is similar to an autoencoder. It learns an LSTM as the encoder that maps the sequential data to a latent representation in a probabilistic manner, and decodes the latents back to data.

4.2 Performance Evaluation

To answer **Q1**, we apply DTS to domain adaptation tasks to validate the effectiveness of our proposed disentanglement strategy, and compare DTS with the state-of-the-art algorithms. During the training phase, only the training dataset is accessible: labeled data as the source domain and unlabeled data as the target domain. During the evaluation, the test data becomes available as the target domain.

4.2.1 Quantitative Results. Table 1 compares the performance of DTS and the baselines on HAR, HHAR, WISDM AR, and uWave datasets. We include no adaptation as an approximate lower bound, and models trained directly on labeled target data as the upper bound. After the group disentanglement, two variables are used to encode the domain information and the class information, respectively. We use the class-dependent latent variable for classification since it is domain-invariant. We observe that DTS outperforms the baselines with consistently +3% higher accuracy over all datasets. These results ascertain the effectiveness of DTS in boosting the performance of domain adaptation by obtaining domain-invariant transferable components as common knowledge. The enhanced results also validate eliminating irrelevant information from group disentanglement could prevent negative transferring.

4.2.2 Ablation Studies. We study whether DTS can decompose the representations into domain-dependent and class-dependent components with a series of ablation studies. We compare the discriminability of the disentangled features, including domain-dependent (g_d) and class-dependent (g_y) components (see Section 2.4.2), sampled from both source and target domains. The ablations include (i) using domain-dependent features g_d and class-dependent features g_y, and (ii) domain-invariant and class-invariant features (shown as /r), respectively. The comparison between DTS and the ablations (/r) is shown in Table 2. We observe that DTS significantly outperforms the ablations over all datasets. Disentangled task-dependent group segments consistently help to improve the performance. Conversely, the class-dependent features are invariant to the change of domains, and the domain-dependent features are invariant to the change of classes. DTS could preserve the discriminability of the disentangled features corresponding to the specified task, and simultaneously make disentangled features less discriminative for the irrelevant task. It indicates that these disentangled group segments do not contain any useful semantic concepts for other irrelevant tasks.

4.3 Individual Latent Factor Disentanglement

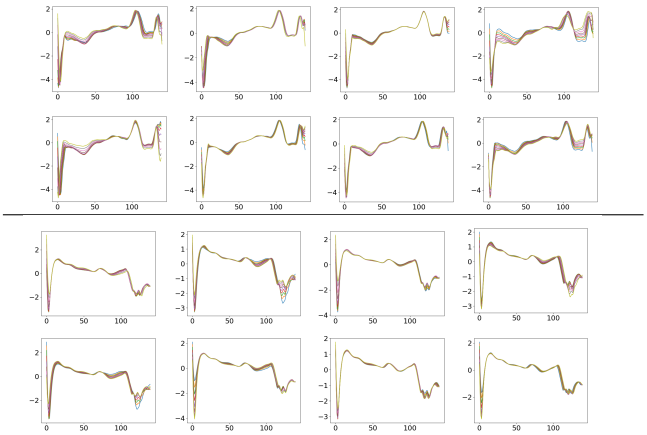
To answer **Q2**, we provide latent traversals as qualitative results to validate that DTS tends to consistently discover more informative latent factors and provide more meaningful disentangled representations of time series (see Section 2.3).

4.3.1 Traversal Plots to Discover Semantics. There is currently no general method for quantifying the degree of learned disentanglement or optimizing the hyperparameters (unless there are concept ground-truth factors v available, then the MI gap [8, 28] could be used to determine if there exists a deterministic, invertible relationship between z and v). Thus, we can not quantitatively compare the degree of disentanglement achieved by different models or when optimizing the hyperparameters of a single model. Fig. 4 plots the

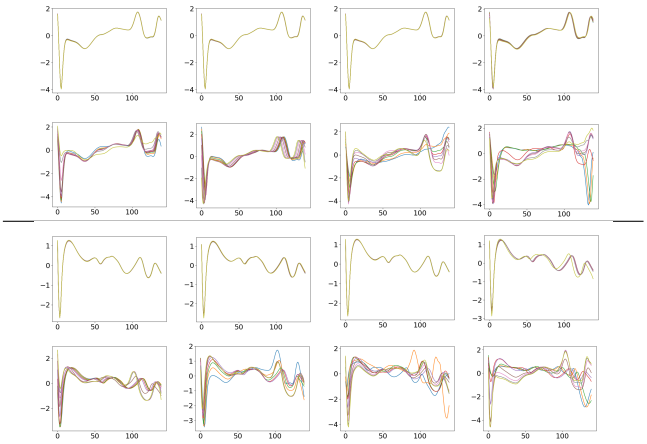
manipulation results of the latent traversal results from DTS. Each block of the figure corresponds to the traversal of a single latent variable while keeping others fixed. Each row represents a different seed image used to infer the latent values with traversal over the [-4, 4] range. The results show that our manipulation approach performs well on all attributes in both positive and negative directions. We observe that moving the latent variables can produce continuous change, with the sequential patterns orthogonal to the others. According to the editing process, the first two rows (sampled from two different time series of ECG) denote the decline degree at the first turning point, transition from rigid to a more mild manner; the last two rows (with the same sampling strategy) denote the rising trend at the second turning point, transition from inconspicuous to obvious. It shows that DTS discovers latent factors that encode sequential trends and depict an interpretable property in the generation. These observations provide strong evidence that DTS does not produce time series randomly but learns some interpretable semantics in the latent space.

4.3.2 Qualitative Comparison of Latent Variables. We train DTS on ECG data to evaluate disentanglement performance for individual latent factors. We use the same traversal way to show the disentanglement quality. Fig. 5 provides a qualitative comparison of the disentanglement performance of DTS and LSTM-VAE. We edit the time series by altering the latent variables in the $\mathcal{Z}$ space. Here, the dimension of the representation is set to be 12. This setting helps reduce the impact of differences in complexity by model frameworks. However, for a better comparison, we only select eight dimensions that change more regularly. The sequences visualized in panels are generated from $Z \sim q(Z \mid \boldsymbol{x}_{1:T})$. Hence, the dynamics are imposed by the encoder, but the identity is sampled from the prior.

Fig. 5 shows traversals in latent variables that depict an interpretable property in generating time series. Often it could generate more semantic convincing time series than LSTM-VAE. It can be seen that sampling from an entangled representation results in LSTM-VAE (panel (a)) only reflects small differences according to the traversal perturbations. One possible reason is that the LSTM-VAE is dominated by the reconstruction term The slight changes only correspond to the reconstruction distortion due to the latent variables and observations are independent. Comparing with LSTM-VAE, some of the DTS latent variables tend to learn a smooth continuous transformation over a wider range of factor values as vibrations. A clear transition process can be observed from the manipulation results with respect to the $\mathcal{Z}$ space. As the value of individual latent factor increases, the semantic of the latent factor changes across different sequential patterns. Other latent variables are robust with the vibrations, as it does not play any role in the generation process. Single latent units are sensitive to changes in single generative factor, while being relatively invariant to changes in other factors. One possible reason is that the representation of the time series could be effectively expressed with a few latent variables in the $\mathcal{Z}$. Individual latent factor disentanglement process may help us to recognize the useful latent variables, and discard the redundant parts. All these results demonstrate that DTS is able to disentangle useful knowledge from sequential data, which is more informative as interpretable factors in the latent space.



(a) Visualizing the generated time series from LSTM-VAE.

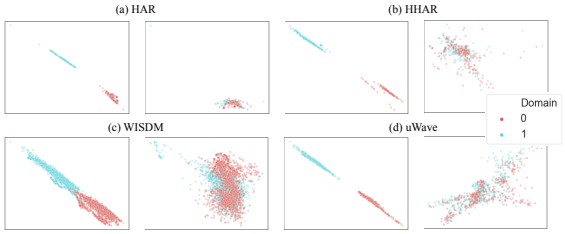


(b) Visualizing the generated time series from DTS.

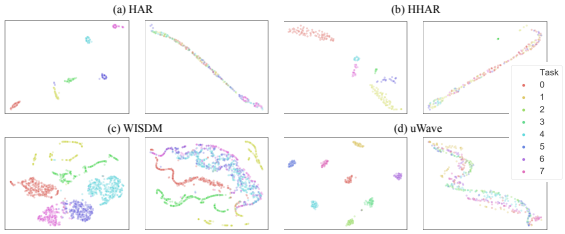
Figure 5: Comparison of learned latent variables. Traversals depict an interpretable property in generating time series from eight-dimensions of the latent variables Z (shown as 8 subfigures in a group). Traversals are sampled from two different time series of ECG (separated by the black line), with range [-4,4] (shown as 9 lines in one subfigure).

4.4 Latent Group Segment Disentanglement

To answer **Q3**, we visualize the disentangled segments in the representation space (see Section 2.4.1). Fig. 6(a) shows the effect of domain-dependent and domain-invariant disentanglement on the distribution of the extracted features. For all the datasets, the adaptation in DTS makes the disentangled domain(class)-dependent features more distinguishable, but the domain(class)-invariant features indistinguishable. The results validate that DTS can learn decomposed segments that contain independent semantic information. Further, we can observe an apparent clustering effect (the different colors denote different categories). A widely-accepted assumption [5] indicates that observation distribution contains separated data clusters and data samples in the same cluster share the same class label in domain adaptations. These results validate the discriminative ability of the disentanglement. It almost matches the prior perfectly, as the semantically similar observations are mapped



(a) T-SNE visualizations of the DTS activations on the distribution of domain-dependent representation (left) and domain-invariant representations(right). Blue points correspond to the source domain examples, while red ones correspond to the target domain ones.



(b) T-SNE visualizations of DTS activations on the distribution of class-dependent (left) and class-invariant (right) representations. Each color denotes one specific class.

Figure 6: The effect of (a) domain-dependent and -invariant (b) class-dependent and -invariant disentanglement on the distribution of the extracted features. In all cases, the adaptation in DTS makes the dependent/invariant features from different sources more/less distinguishable, respectively.

closer, and create clusters. This phenomenon suggests that the disentangled group segments could enhance the interpretability.

5 CONCLUSION

In this paper, we investigated a novel and challenging problem of learning disentangled time series representations. We proposed DTS, a multi-level disentanglement approach, covering both individual latent factor and group semantic segments, to generate hierarchical semantic concepts as the interpretable and disentangled representation. DTS can balance the preference between correct inference and fitting data distribution. It also alleviates the KL vanishing problem by introducing a mutual information maximization term while preserving a heavier penalty on the dimension-wise KL to keep the disentanglement property. The experimental results on five benchmark datasets demonstrated the effectiveness of DTS in learning interpretable semantic concepts with disentanglement.

ACKNOWLEDGMENTS

The work is done during an internship at NEC, and is in part, supported by NSF (IIS-1750074, CNS-1816497). The views and conclusions in this paper are those of the authors and should not be interpreted as representing any funding agencies.

REFERENCES

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. 2014. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446* (2014).
[2] Davide Anguita and *et al.* 2013. A public domain dataset for human activity recognition using smartphones.. In *Esann*.
[3] Shaojie Bai et al. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271* (2018).
[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* (2010).
[5] Shai Ben-David and Ruth Urner. 2014. Domain adaptation–can quantity compensate for quality? *Annals of Mathematics and Artificial Intelligence* (2014).
[6] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, et al. 2018. Understanding disentangling in beta-VAE. *arXiv:1804.03599* (2018).
[7] Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. 2019. Learning disentangled semantic representation for domain adaptation. In *IJCAI*.
[8] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. 2019. Isolating Sources of Disentanglement in VAEs. In *NeurIPS*.
[9] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Variational lossy autoencoder. *arXiv:1611.02731* (2016).
[10] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. 2015. A recurrent latent variable model for sequential data. *arXiv preprint arXiv:1506.02216* (2015).
[11] Vincent Fortuin and *et al.* 2018. SOM-VAE: Interpretable discrete representation learning on time series. *arXiv preprint arXiv:1806.02199* (2018).
[12] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*.
[13] Xiaojie Guo and et al. 2020. Interpretable Deep Graph Generation with Node-Edge Co-Disentanglement. In *KDD*.
[14] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework. (2016).
[15] Geoffrey E Hinton and Richard S Zemel. 1994. Autoencoders, minimum description length, and Helmholtz free energy. *NeurIPS* (1994).
[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* (1997).
[17] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
[18] Jennifer R Kwapisz and *et al.* 2011. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter* (2011).
[19] Yuening Li, Zhengzhang Chen, Daochen Zha, Kaixiong Zhou, Haifeng Jin, Haifeng Chen, and Xia Hu. 2021. Automated Anomaly Detection via Curiosity-Guided Search and Self-Imitation Learning. *IEEE Transactions on Neural Networks and Learning Systems* (2021).
[20] Yuening Li, Zhengzhang Chen, Daochen Zha, Kaixiong Zhou, Haifeng Jin, Haifeng Chen, and Xia Hu. 2021. Autood: Neural architecture search for outlier detection. In *International Conference on Data Engineering (ICDE)*.
[21] Yuening Li, Xiao Huang, Jundong Li, Mengnan Du, and Na Zou. 2019. Specae: Spectral autoencoder for anomaly detection in attributed networks. In *International Conference on Information and Knowledge Management*.
[22] Jiayang Liu and *et al.* 2009. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing* (2009).
[23] Chen Luo, Zhengzhang Chen, Lu-An Tang, Anshumali Shrivastava, Zhichun Li, Haifeng Chen, and Jieping Ye. 2018. TINET: Learning Invariant Networks via Knowledge Transfer. In *KDD*.
[24] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644* (2015).
[25] Arthur J Moss and *et al.* 1995. ECG T-wave patterns in genetically distinct forms of the hereditary long QT syndrome. *Circulation* (1995).
[26] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. 2017. Variational Recurrent Adversarial Deep Domain Adaptation.. In *ICLR*.
[27] Stanislau Semeniuta and *et al.* 2017. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390* (2017).
[28] Huajie Shao and *et al.* 2020. Controlvae: Controllable variational autoencoder. In *ICML*.
[29] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. 2020. Interfacegan: Interpreting the disentangled face representation learned by gans. *TPAMI* (2020).
[30] Allan Stisen and *et al.* 2015. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *SenSys*.
[31] Eric Tzeng and *et al.* 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).
[32] Prince Zizhuang Wang and William Yang Wang. 2019. Riemannian normalizing flow on variational wasserstein autoencoder for text modeling. *arXiv preprint arXiv:1904.02399* (2019).
[33] Zirui Wang and *et al.* 2019. Characterizing and avoiding negative transfer. In *CVPR*.
[34] Satosi Watanabe. 1960. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development* (1960).
[35] Garrett Wilson and *et al.* 2020. Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *KDD*.
[36] Shengjia Zhao, Jiaming Song, and Stefano Ermon. 2019. Infovae: Balancing learning and inference in variational autoencoders. In *AAAI*.