# Variational Information Bottleneck with Gaussian Processes for Time-Series Classification

Itamar Efrati
*School of Computer science*
*Reichman University*
Herzliya, Israel
itamar.efrati@post.runi.ac.il

Shai Fine
*The Data Science Institute*
*Reichman University*
Herzliya, Israel
shai.fine@.runi.ac.il

***Abstract*—Time series classification problems are prevalent across various domains, often characterized by intra-series relationships within features, and inter-series relationships between the same features over time. Developing a generalized model capable of capturing these intricate properties poses a considerable challenge. In this research paper, we introduce an innovative approach termed Gaussian Process Variational Information Bottleneck (GP-VIB). This model is designed as an end-to-end system, with a primary focus on acquiring a concise representation of the initial sequence. It aims to retain essential information vital for accurate classification.**

**Through our experiments, we illustrate that the proposed GP-VIB model outperforms existing methods on renowned benchmark datasets.**

***Index Terms*—Information Bottleneck, Variational Inference, Time Series Classification**

## I. Introduction

Time-series data is characterized by inter-series relations between inferred features and intra-series relations within every such feature. The challenge of finding and understanding these features and their relations over time is motivated by applications in finance, healthcare, signal processing and other fields, where understanding and categorizing temporal patterns is crucial for Time-Series Classification (TSC).

Common approaches to TSC involve leveraging diverse techniques such as distance-based methods, feature engineering-based methods, and hybrid methods. More advanced methods include the use of deep learning models, recurrent neural networks (RNNs), and convolutional neural networks (CNNs) tailored to handle the sequential nature of time series data.

An alternative approach for classification places a central focus on constructing a meaningful representation by transforming the original data into a distinct space conducive to the classification process. In classical machine learning, Principal Component Analysis (PCA) employs a linear transformation for this purpose, while more contemporary approaches employ non-linear transformations, such as deep Auto-Encoders (AE), and later on, Variational Auto-Encoders (VAE) [20]. Unlike the deterministic nature of the AE, VAE are stochastic models that learn the distribution of the latent space, which, by capturing latent space characteristics allow them to demonstrate increased robustness and superior performance, essential for reconstructing input data and generating new samples.

As for time-series data, VAE, with it's mean field assumption, may not be well-suited for time-series data due to the challenge of capturing temporal dependencies effectively within the assumed factorized latent variable distributions. A related model, more suitable for time-series data, is the Gaussian Process Variational Auto-Encoder (GP-VAE) [13], which, instead of the mean field assumption, utilizes deep variational inference with a Gaussian process prior, enabling the acquisition of a structural variational posterior that adeptly captures intricate temporal dynamics in time-series. Originally introduced as a method for imputing missing values in time-series data, the GP-VAE model can also be applied to model complete data distributions. Leveraging the approximated posterior, latent representations can be generated and subsequently employed as input for a classifier.

However, questions arise regarding the information retained in the GP-VAE's latent space, what type of data might be excluded, and how to ensure that the compressed data version preserves relevant information for the classification task. It is crucial to ascertain that the model does not omit crucial information for classification and to address potential disparities between data relevant for reconstruction and that pertinent to classification. Thorough evaluation and consideration of these aspects are essential to guarantee the efficacy of the compressed data in supporting the classification task.

We introduce the **G**aussian **P**rocess **V**ariational **I**nformation **B**ottleneck (GP-VIB), a novel deep learning probabilistic model tailored for TSC. By seamlessly integrating the Variational Information Bottleneck (VIB) method [1] with a GP prior, our model excels in acquiring a discriminative latent space that effectively captures the essential temporal dynamics crucial for classification tasks. To enhance our model's capacity for feature extraction and parameter learning, we incorporate InceptionTime blocks [18] into the encoder architecture. This inclusion enables our GP-VIB model to efficiently process time-series data, discern intricate temporal patterns, and derive comprehensive representations. Unlike the GP-VAE model, which entails two distinct steps for classification, our proposed GP-VIB model offers an end-to-end solution, streamlining the classification process for improved efficiency

and performance. Furthermore, our model is on par with state-of-the-art (SOTA) models on benchmark datasets for TSC.

The rest of the paper is organized as follows. We start by presenting related work and providing some essential background for the methods that we use. Next, we present the GP-VIB model in more details. We then move to describe a set of experiments designed to evaluate the utility of the GP-VIB model. Our experiments include comparisons between the GP-VIB and GP-VAE model over the Healing MNIST [21]. Additionally, we present comparisons with SOTA models over the UEA archive [2] for multivariate time-series datasets. Finally, we conclude with a succinct summary of our results and offer suggestions for future research directions.

## II. Background and Related Work

In what follows, we review the different approaches for time-series classification, and provide essential background for the derivation of our suggested method.

### A. Time-Series Classification

In recent years, the field of TSC has experienced intensive activity, with numerous methods and approaches developed for classification. These methods can be broadly categorized to classical machine learning and deep learning. In the following, we begin by discussing machine learning methods, alongside their evolution. This is then followed by discussing a specific deep learning method pertinent to the proposed GP-VIB model: the InceptionTime model.

*1) Classical Machine Learning:* Classical machine learning methods can be further categorized into three main groups: direct classification methods, feature engineering-based methods, and hybrid methods.

Within the realm of direct classification methods, there exist two notable subgroups **distance-based** methods and **interval-based** methods. Distance-based methods involve quantifying dissimilarities between time-series and utilizing these measures for classification. Historically, distance measures were integrated with the K-Nearest-Neighbors (K-NN) algorithm, with the 1-NN Dynamic Time Warping (DTW) standing out as a noteworthy approach [29]. A recent advancement in this domain is the ProximityForest ensemble [23], which employs a combination of 11 different distance functions and Proximity Tree classifiers.

In contrast, **interval-based** methods entail dividing time-series into random intervals and subjecting them to processing by a random forest, where each tree employs distinct splits of the input data. An extension of this concept is the Randomised Supervised Time Series Forest [5], a method that markedly enhances previous methods by introducing randomness in the positioning of intervals within time-series.

**Feature engineering-based** methods involve creating complex features from time-series data and feeding them into traditional classifiers like linear models or random forests. Various assumptions about the data have led to different types of feature engineering methods. A notable example is TSFRESH [8], which extracts 800 features from time-series, followed by feature selection before classification. A more advanced approach is FreshPRINCE [24] that utilizes a rotation forest classifier over the features, without performing feature selection.

Another approach involves **kernel-based** methods that generate a large number of kernels to create features for simpler classifiers. A well-known model is ROCKET [9], which generates thousands of convolutional kernels, applies pooling operations, and employs these outputs as features in a Ridge regression classifier. Further extensions of ROCKET that combine additional types of kernels are the Multi-ROCKET [33] and Hydra-MultiROCKET (Hydra-MR) [10].

**Shapelets** are discriminative subsequences that can indicate a class within time-series datasets. The Random Dilated Shapelet Transform [17] randomly selects numerous shapelets from the training data and uses features derived from these shapelets to train a linear RIDGE classifier.

An alternative approach is **dictionary-based**. It treats phase-independent subsequences as words using techniques like Symbolic Fourier Approximation. These methods record the frequency of repeating patterns to create features. Algorithms like WEASEL-D [30] and TDE [25] fall under this category.

**Hybrid** methods combine various aforementioned approaches. An notable exemple is HIVE-COTE v2.0 (HC2) [26], an ensemble method that integrates multiple techniques. This approach stands as the state-of-the-art solution across numerous benchmark datasets.

*2) InceptionTime:* Inception [32] is an image classification neural network model. Its unique architecture is constructed by a concatenation of different kernels over the image, rather than a cascade architecture. It was later upgraded to Inception-v4 [31] by adding a residual connection, which improves its performance. The InceptionTime model [18] adapts this image classification architecture to time-series classification. It uses a 1-dimensional CNN layers, sets bigger kernels than the kernel of the image classification model, and adds a "bottleneck" module in each Inception block in order to control the input size throughout the network (cf. Fig 1).

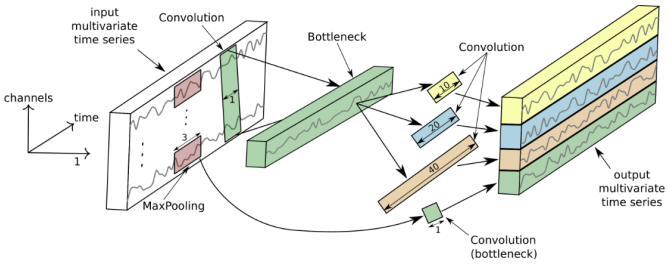



Fig. 1: Single InceptionTime block. Source [18]

### B. Deep Variational Information Bottleneck

The Information Bottleneck (IB) method is an Information Theoretic technique, introduced by Tishby et al. [34].

The main idea behind the IB method is to find a (compressed) representation of the input data that maximizes the mutual information with the relevant output (task) while minimizing the mutual information with the irrelevant parts of the input. More formally, given a random variable $\boldsymbol{X}$ representing the input data and a random variable $\boldsymbol{Y}$ representing the output, the IB optimization problem can be expressed as:

$$\max I(\boldsymbol{Z}, \boldsymbol{Y}) \text{ s.t. } I(\boldsymbol{X}, \boldsymbol{Z}) \leq I_c \tag{1}$$

where $I_c$ is an information constraint. Equivalently, by introducing a Lagrange multiplier $\beta$, the IB objective function can be stated as follows:

$$R_{IB} = I(\boldsymbol{Z}, \boldsymbol{Y}) - \beta \cdot I(\boldsymbol{X}, \boldsymbol{Z}) \tag{2}$$

where:

$$I\left(\boldsymbol{Z}, \boldsymbol{Y}\right) = \int p(z, y) \log \frac{p(y|z)}{p(y)} \, dy \, dz \tag{3}$$

$$I\left(\boldsymbol{X}, \boldsymbol{Z}\right) = \int p(x, z) \log p(z|x) \, dz \, dx - \int p(z) \log p(z) dz \tag{4}$$

Here, $\boldsymbol{Z}$ is the learned lattent representation, and $\beta$ is a trade-off parameter. $I(\boldsymbol{Z}, \boldsymbol{Y})$, represents the mutual information between the representation and the output, while the the term $\beta \cdot I(\boldsymbol{X}, \boldsymbol{Z})$, represents the mutual information between the input data and the lattent representation, weighted by $\beta$.

The three random variables $\boldsymbol{X}, \boldsymbol{Z}$ and $\boldsymbol{Y}$ form a Markov chain: $\boldsymbol{Y} \leftrightarrow \boldsymbol{X} \leftrightarrow \boldsymbol{Z}$. By this assumption we get that $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are $d$-separated given $\boldsymbol{X}$, i.e. conditionally independent, such that $p\left(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y}\right) = p\left(\boldsymbol{Z}|\boldsymbol{X}\right)$. This leads to the following factorization of the joint distribution:

$$p\left(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}\right) = p\left(\boldsymbol{Z}|\boldsymbol{X}\right) p\left(\boldsymbol{Y}|\boldsymbol{X}\right) p\left(\boldsymbol{X}\right) \tag{5}$$

Calculating both mutual information measurements is intractable. Alemi et al. [1] suggested both an upper and a lower bound for the mutual information using variational inference. Specifically, Eq. (4) requires calculating $p\left(z\right) = \int p\left(z|x\right) p\left(x\right) \, dx$, which is intractable because it involves calculating the evidence $p\left(x\right)$. Hence, $r\left(z\right)$ is introduced as a variational approximation for $p\left(z\right)$. Since: [1]

$$KL\left[p\left(\boldsymbol{Z}\right) || r\left(\boldsymbol{Z}\right)\right] \geq 0 \rightarrow \int p\left(z\right) \log p\left(z\right) dz \geq \int p\left(z\right) \log r\left(z\right) dz \tag{6}$$

we get the following upper bound:

$$\begin{aligned} I\left(\boldsymbol{X}, \boldsymbol{Z}\right) &\leq \int p(x, z) \log p(z|x) \, dx \, dz - \int p(z) \log r(z) dz \\ &= KL\left[p\left(\boldsymbol{Z}|\boldsymbol{X}\right) || r(\boldsymbol{Z})\right] \end{aligned} \tag{7}$$

[1] $KL\left[p||r\right]$ is the Kullback-Leibler Divergence measure:

$$KL\left[p(\boldsymbol{Z})||r(\boldsymbol{Z})\right] = \int p(z) \log \frac{p(z)}{r(z)} \, dz$$

As for Eq. (3), the calculation requires calculating $p\left(y|z\right)$. By applying the Markov chain assumption, we get that:

$$p\left(y|z\right) = \int p\left(x, y|z\right) dx = \int \frac{p\left(y|x\right) p\left(z|x\right) p\left(x\right)}{p\left(z\right)} dx \tag{8}$$

Eq. (8) is intractable since the above requires calculating $p\left(x\right)$. By introducing the variational approximation $q\left(y|z\right)$, we get that

$$KL\left[p(\boldsymbol{Y}|\boldsymbol{Z})||q(\boldsymbol{Y}|\boldsymbol{Z})\right] \geq 0 \rightarrow \int p\left(y|z\right) \log p(y|z) dy \geq \int p\left(y|z\right) \log q(y|z) dy \tag{9}$$

which leads to the following lower bound:

$$\begin{aligned} I\left(\boldsymbol{Z}, \boldsymbol{Y}\right) &\geq \int p(y, z) \log \frac{q_(y|z)}{p(y)} \, dy \, dz \\ &= \int p(y, z) \log q_(y|z) \, dy \, dz + H(Y) \end{aligned} \tag{10}$$

$H(\boldsymbol{Y})$ (the entropy of $\boldsymbol{Y}$) is a positive constant and therefore it can be ignored in the optimization problem. As for the other term, by writing the marginalization form of $p(y, z)$ and by applying the Markov chain assumption, it follows that:

$$p(y, z) = \int p(y, z, x) dx = \int p(x) p(y|x) p(z|x) dx$$

This leads to the following lower bound:

$$I\left(\boldsymbol{Z}, \boldsymbol{Y}\right) \geq \int p(x) p(y|x) p(z|x) \log q(y|z) \, dx \, dy \, dz \tag{11}$$

Putting it all together, the Variational Information Bottleneck (VIB) objective function follows:

$$J_{VIB} = \frac{1}{N} \sum_{n=1}^{N} \Big[ \int p\left(z|x_n\right) \log q\left(y_n|z\right) dz - \beta \cdot KL\left[p\left(z|x_n\right) || r\left(z\right)\right] \Big] \tag{12}$$

### C. Structured Variational Inference

In their paper, Kingma and Welling [20] set the variational distribution to be a diagonal covariance (independent) multivariate Gaussian, i.e. a mean field approximation. Moving to a structured variational inference, the variational distribution is represented by a full covariance multivariate Gaussian distribution. This results with a reparameterization gradient that involves a dense matrix multiplication that scales as $O(T^2)$, where $T$ is the number of time steps. Bamler and Mandt [3] suggested an algorithm for the reparameterization gradient in $O(T)$. The main idea is to infer the parameters of the full covariance matrix by its precision matrix. Using the Cholesky decomposition, the precision matrix $\boldsymbol{\Lambda}$ can be written as the product of a bidiagonal matrix and its transpose:

$$\boldsymbol{\Lambda} := \boldsymbol{B}^T \boldsymbol{B}, \, with \, \boldsymbol{B} = \begin{cases} b_{tt'}, & \text{if } t' \in \{t, t+1\} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

This parameterization leads to $\mathbf{\Lambda}$ being positive definite, symmetric and tridiagonal. While the precision matrix is sparse, it's inverse, the covariance matrix, can be dense, which in turn enables the use of a broader and richer set of variational distribution than the set of distributions resulted from the mean field approximation. Moreover, learning the $B$ matrix requires to estimate only $2T-1$ parameters, i.e. it scales as $O(T)$ similarly to the mean field effort.

### D. GP-VAE

Multivariate time-series with missing values are quite frequent in real life settings. Classical time-series imputation methods do not take into account the complex interactions between the different channels. In addition, many time-series methods fail to model data with missing values. Fortuin et al. suggested a VAE with a Gaussian Process (GP) prior and a structure variational posterior, termed GP-VAE [13]. This model can learn a latent representation of input time-series with missing data that captures complex dynamics between different channels in the data. In addition, the model is able to reconstruct an imputed version of the input time-series. GP-VAE uses the structured inference algorithm [3] in order to learn a variationl posterior for each feature in the latent space, i.e. $q(\boldsymbol{z}_{1:T,j}|\boldsymbol{x^o}_{1:T}) = N(m_j, \Lambda_j^{-1})$, where $j$ is the index of the feature in the latent space and $\boldsymbol{x^o}$ is the input time-series with only observed values. The prior of $\boldsymbol{Z}$ is represented as a GP such that $z(j) \sim GP(m_z(\cdot), k_z(\cdot,\cdot))$, where $k_z$ is the Cauchy kernel, which is appropriate for modelling data that varies at multiple time scales:

$$K_{Cau}(\tau,\tau') = \sigma^2\left(1+\frac{(\tau-\tau')^2}{l^2}\right)^{-1} \tag{14}$$

In order to learn only from the observed data, they devised an evidence lower bound (ELBO) such that its reconstruction error term sums only over the observed values, reminiscent of the HI-VAE method [28]. The model's ELBO follows:

$$\sum_{t=1}^{T} E_{q(\boldsymbol{z}_t|\boldsymbol{x}_{1:T})}\left[\log p\left(\boldsymbol{x}_t^o|\boldsymbol{z}_t\right)\right] - \beta \cdot D_{KL}\left[q\left(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}\right)||p\left(\boldsymbol{z}_{1:T}\right)\right] \tag{15}$$

## III. The GP-VIB Model

We suggest a novel Variational Information Bottleneck model that can handle time-series data. In this section, we define the prior distribution of $\boldsymbol{Z}$, the encoder, the decoder, and specify the GP-VIB objective, similarly to Eq. (12).

### A. Problem Setting

Let $\boldsymbol{X} \in \mathbb{R}^{T\times d}$ denote a multivariate time-series of length $T$, where each instance in the sequence is a vector of size $d$ features such that $\boldsymbol{x_t} = [x_{t1},\ldots,x_{td}] \in \mathbb{R}^d$. For each time-series there is a a single label $y \in \boldsymbol{Y}$, where $\boldsymbol{Y}$ is a finite set of categories.

Hence, our modelling approach suggests to construct a (latent) dense time-series representation $\boldsymbol{Z} \in \mathbb{R}^{S\times d'}$ of the input time-series $\boldsymbol{X}$ that preserves the relevant information for the classification task, i.e. predicting $y$. Each time-series in the latent space consists of $S$ data points, where each instance in the sequence is a vector of size $d'$ such that, $\boldsymbol{z_s} = [z_{s1},\ldots,z_{sd'}]$

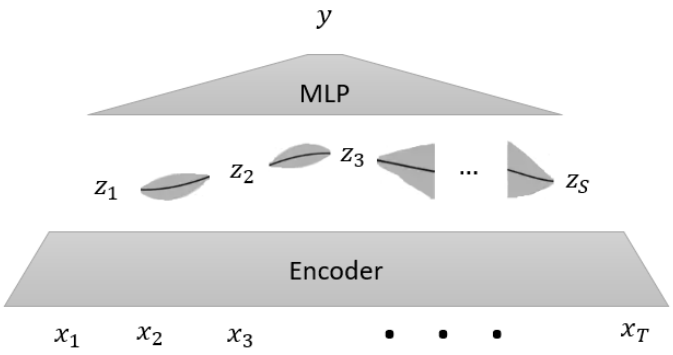



Fig. 2: Overview of the GP-VIB architecture. The encoder is comprised of InceptionTime blocks and a compression block. The latent sequence, z, is the input to the MLP decoder, which in turn predicts $y$.

### B. A Graphical Model Perspective

The IB principle states that random variables $\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{Z}$ form a Markov chain. In the current setting, it should be noted that $\boldsymbol{X}$ and $\boldsymbol{Z}$ are time-series of different lengths. This leads to the graphical model depicted in Fig. 3.

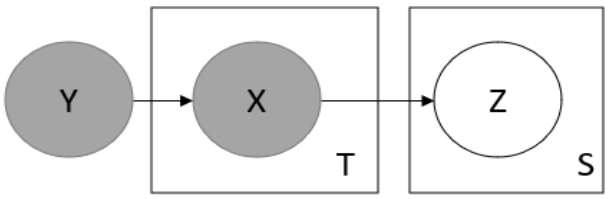



Fig. 3: A graphical model illustration of the GP-VIB model. $\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{Z}$ form a Markov chain, while $\boldsymbol{X}$ and $\boldsymbol{Z}$ are time-series of lengths $T$ and $S$, reps.

The graphical model perspective suggests that $\boldsymbol{Z_n}$ and $y_n$ are $d$-separated given $\boldsymbol{X_n}$ namely $y_n$ and $\boldsymbol{Z_n}$ are independent. Hence, the GP-VIB model architecture consists of two parts:

- The encoder (inference model): given the input $\boldsymbol{X_n}$ the encoder models the posterior distribution $p\left(\boldsymbol{Z_n}|\boldsymbol{X_n}\right)$.
- The decoder (prediction model): given the latent representation the decoder models $p\left(y_n|\boldsymbol{Z_n}\right)$.

Similar to their role in the VIB model, both the encoder and the decoder are implemented using deep neural networks.

### C. The Prior Distribution of $\boldsymbol{Z}$

The definition of the prior $r(z)$ is an important step in variational inference. We would like to choose a prior that reflects the time dynamics and relationships between features over time. Following [13], we set the prior to be a GP, specifically a Multivariate Gaussian with mean $\mathbf{0}$ and covariance matrix constructed by a Cauchy kernel. We further assume that every feature in the latent space is a GP, i.e., $z_{1:T,j} \sim GP\left(m_z\left(\cdot\right), k_z\left(\cdot,\cdot\right)\right)$, where all features in the latent space share the same prior.

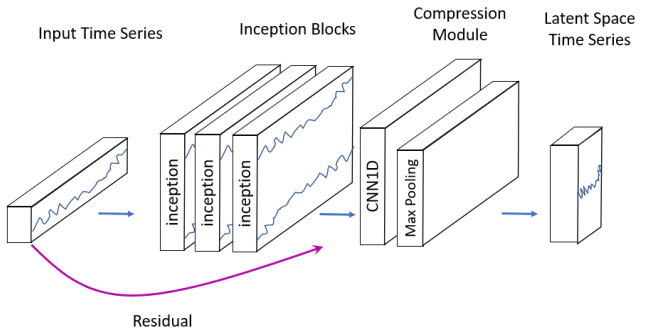



Fig. 4: GP-VIB encoder overview. The encoder is comprised of 3 InceptionTime modules followed by a residual module. Then the compression block compresses the time-series into the desired latent space size.

### D. The Encoder

The encoder models the posterior distribution, $p\left(\boldsymbol{Z_n}|\boldsymbol{X_n}\right)$, where $\boldsymbol{X_n}$ is a time-series. We use a neural network to approximate the parameters of the variational distribution. Fig. 4 illustrates the complete architecture of the encoder. We use 3 InceptionTime blocks (Fig. 1) with a residual layer followed by a compression block, which compresses the time-series to the desired latent space dimension. To set the number of channels in the time-series we use a 1-dimensional CNN layer, where the number of output filters are the desired channels in the latent time-series. In order to set the length of the latent time-series, we use an adaptive max pooling layer.

Next, we apply the structured variational inference algorithm and extract the mean and the covariance matrix of the latent representation. This leads to the following posterior:

$$p\left(\boldsymbol{Z_n}|\boldsymbol{X_n}\right) = p\left(\boldsymbol{Z_n}|f_\psi\left(\boldsymbol{X_n}\right)\right) \tag{16}$$

where $f_\psi$ is a neural networks.

### E. The Decoder

The decoder, $q\left(y_n|\boldsymbol{Z_n}\right)$, is a multinomial distribution, conditioned on the latent series $\boldsymbol{Z_n}$, such that:

$$q\left(y_n|\boldsymbol{Z_n}\right) = q\left(y_n|g_\theta\left(\boldsymbol{Z_n}\right)\right) \tag{17}$$

where $g_\theta\left(\cdot\right)$ is a nonlinear function with the parameters $\theta$ that outputs the logits of the multinomial distribution. Thus, the log-likelihood expression $\log q\left(y_n|g_\theta\left(z_{1:T}\right)\right)$ is essentially the cross-entropy loss function. As for $g_\theta\left(\cdot\right)$, we use a "Flattened" decoder, which takes the latent time-series and flattens it into one vector. This vector serves as an input to the Multi-Layer Perceptron, which outputs the logits of the label.

### F. The ELBO

Putting it all together, the ELBO of the GP-VIB model follows:

$$\begin{aligned} L = \frac{1}{N}\sum_{n=1}^{N} & \mathbb{E}_{p(\boldsymbol{Z}_n|f_\psi(\boldsymbol{X}_n))}\log q\left(y_n|g_\theta\left(\boldsymbol{Z}_n\right)\right) \\ & - \beta KL\left[p\left(\boldsymbol{Z}_n|f_\psi\left(\boldsymbol{X}_n\right)\right)||r\left(\boldsymbol{Z}_n\right)\right] \end{aligned} \tag{18}$$

## IV. Experiments

In this section we present two sets of experiments designed to evaluate and compare performance[2]: Firstly, we compare the performance of GP-VIB vs. GP-VAE and other VAE architectures. Secondly, we perform a comparative evaluation of GP-VIB vs. SOTA models specifically designed for TSC.

Similar to other VAE models, the computational efficiency of our VIB model with a multivariate Gaussian prior, such as a GP, is primarily influenced by the design of the encoder and decoder architectures. The choice of a Gaussian Process as a prior does not introduce significant additional complexities compared to standard VAE models. Thus, the main factors affecting computational efficiency are consistent with those found in typical VAE models, depending largely on the architecture and parameterization of the encoder and decoder.

### A. GP-VIB vs. Different VAE Architectures

GP-VAE was suggested as an imputation method for time series data. Hence, we used the Healing MNIST dataset [21], which is designed to reflect attributes of irregularly sampled time series that are common in the healthcare domain. The dataset contains sequences of the same digit that rotates randomly between frames. The analogy to healthcare is drawn by the fact that every frame (an image of a digit) is a collection of measurements (every pixel is a measure), and this presumably describes the state of a patient's health. The frames contain around $60\%$ missing pixels and the rotation between two consecutive frames is normally distributed. We followed the exact experiment setting used in [13]. For comparison, we used VAE [20], HI-VAE [28], GP-VAE [13], Logistic Regression model, VIB [1], and GP-VIB. We then compared three performance measurements: Accuracy, AUCROC, and AUPRC. To evaluate the VAE models, we implemented them using the architectures proposed in [13], by using a logistic regression model on the imputed data for the classification task. In addition we tested the Logistic Regression model over the missing data directly. As for the GP-VIB architecture, we implemented the encoder and the prior that were used by the GP-VAE architecture. To hold a fair comparison between our

[2]Our code can be found at https://anonymous.4open.science/r/GP-VIB-TSC-45E9/README.md

model and the VAE models, we used one linear output layer For the decoder.

For each architecture we train three models with different seeds and report the mean and std of each metric. The comparison is presented in Table I, where evaluation metrics are scaled to 100 for increase readability, and the best results are marked in bold.

TABLE I: Performance over HMnist dataset.

| Model | Accuracy | AUROC | AUPRC |
|---|---|---|---|
| Logistic Regression | $66.6 \pm 0.0$ | $92.2 \pm 0.0$ | $63.5 \pm 0.0$ |
| VAE | $70.0 \pm 2.7$ | $94.2 \pm 0.9$ | $74.2 \pm 0.3$ |
| HI-VAE | $76.6 \pm 0.1$ | $96.6 \pm 0.0$ | $82.8 \pm 0.1$ |
| GP-VAE | $77.9 \pm 0.2$ | $96.5 \pm 0.1$ | $84.1 \pm 0.3$ |
| VIB | $83.0 \pm 0.0$ | $98.1 \pm 0.0$ | $90.0 \pm 0.3$ |
| GP-VIB | $\mathbf{88.2 \pm 0.6}$ | $\mathbf{99.2 \pm 0.0}$ | $\mathbf{95.2 \pm 0.2}$ |

GP-VIB exhibits superior performance over other VAE architectures across multiple metrics, showcasing its efficacy in the context of TSC with missing values. While sharing similarities with GP-VAE, their fundamental objectives differ — The VAE model aims for a latent space conducive to data reconstruction, while the VIB architecture emphasizes a discriminative latent space tailored for downstream tasks. In TSC scenarios with missing values, GP-VAE adopts a two-step process involving imputation before classification. Conversely, GP-VIB excels in learning the dynamics of input time-series data with missing values, creating a compressed version that preserves discriminative information. This unique capability results in an end-to-end model that consistently outperforms others in the various metrics, affirming its effectiveness in handling missing data scenarios.

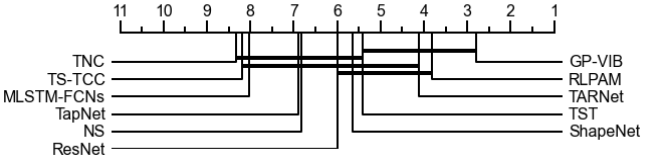
### *B. Performance on the UEA Archive*

We opted to evaluate our model using the UEA [2] benchmark datasets. We excluded datasets featuring series of unequal lengths or missing values, resulting in a selection of 26 out of 30 datasets from the UEA. Following the experimental protocol outlined in [27], we conducted evaluations of the GP-VIB model 30 times with different seeds for each dataset.

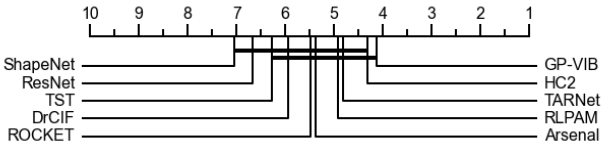
Following recommendations from [11], we utilized the Friedman test to assess the rejection of the null hypothesis. Subsequently, we conducted pairwise post-hoc analysis by [4], replacing the average rank comparison with a Wilcoxon signed-rank test, and applying Holm's alpha (5%) correction [16]. To visually represent this comparison, we employed a critical difference diagram based on the proposal of [11], where a thick horizontal line indicates a cluster of classifiers (a clique) that are not significantly different in terms of accuracy.

During our testing and evaluation, we have experienced difficulties to train the GP-VIB model across datasets in both archives. We found that datasets characterized by a limited training data size are harder to train, which is a challenging scenario for deep learning techniques. To address this issue, we adjusted the latent space size according to the size of the training dataset, utilizing the rational that the larger the training dataset is, the more data we have to approximate the posterior parameters in the latent space. This results in a latent space of size 2 for small datasets (training data with 50 samples), and up to a latent space of size 10 for the largest datasets in the archives.

For this set of experiments, we compared between GP-VIB and the best models presented is [6]. In addition, we also used the results presented at the the tsml-eval github[3] for SOTA models that were missing. For comparison, we added the following deep learning models: ShapeNet [22], time-series Transformer (TST) [36], ResNet, MLSTM-FCNs [19], Negative samples (NS) [14], TNC [35], TS-TCC [12], TAPNet [37], RLPAM [15] and TARNet [7]. As for machine learning models, we added DrCIF and Arsenal, which are components in the HC2 hybrid ensemble model [26] ,and the ROCKET model [9].




(a) GP-VIB vs. Top Deep Learning models




(b) GP-VIB vs. top models

Fig. 5: Accuracy Critical Difference diagrams on UEA archive, for GP-VIB vs. Top Deep Learning models, and GP-VIB vs. combined TSC models.

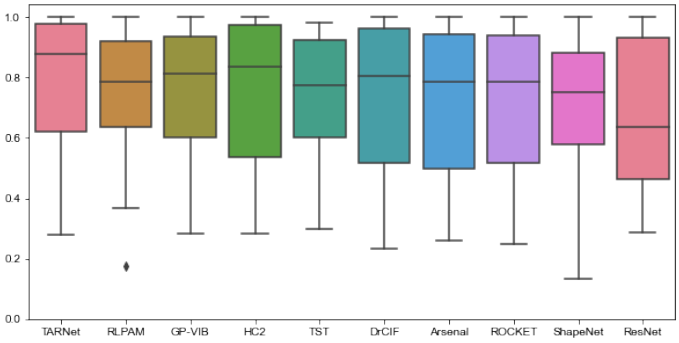
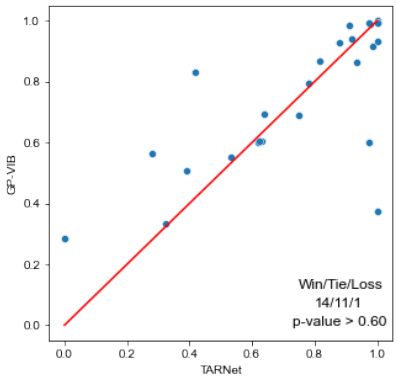


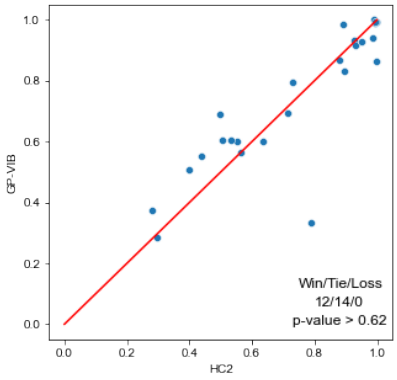

Fig. 6: UEA Test accuracy box-plot, sorted by mean accuracy per model.

Fig. 5a shows that GP-VIB achieves the highest average accuracy rank of the critical different diagrams amongst the deep learning models. Fig. 5b shows that amongst the top models, including machine learning models, GP-VIB is still

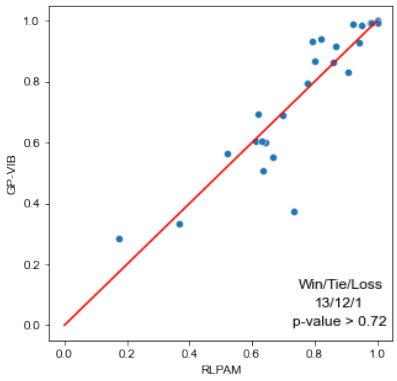
[3]https://github.com/time-series-machine-learning/tsml-eval/tree/main/results/classification/Multivariate




(a) GP-VIB vs. TARNet




(b) GP-VIB vs. HC2




(c) GP-VIB vs. RLPAM

Fig. 7: Pairwise plot of GP-VIB with HC2, RLPAM, and TARNet test accuracy over UEA archive, showing that GP-VIB is comparable to the top models over this archive.

at the top of the critical difference diagram with the best average rank amongst the competitors. In both diagrams, GP-VIB is comparable with the top SOTA models as it is not significantly different from them. While GP-VIB achieve the top rank, Fig. 6 shows that the mean accuracy of GP-VIB is third to TARNet, and RLPAM.

The pairwise plots in Fig. 7 accept the null hypothesis, with a high p-value, confirming that our model presents comparable performance to the 3 top models.

## V. Summary

In this paper we introduce GP-VIB, a novel model specifically crafted for time-series classification, which implements a Variational Information Bottleneck architecture, enriched with a Gaussian Process representation of the latent space. Our model is on par with SOTA models on benchmark datasets for TSC The observed performance underscore the promising potential of GP-VIB in advancing the state of the art in this field.

While VAE typically learns a generative latent space, focused on preserving information necessary for reconstructing the input data, VIB is designed to compel the model to learn a discriminative latent space that encapsulates information crucial for the classification task. By enriching VIB with a GP prior and incorporating structural inference to approximate a structured variational posterior, our approach enables the learning of a latent space that preserves time-dependent dynamics. Our experiments demonstrate that the GP-VIB model performs exceptionally well as a classifier.

A common scenario with real-world data involves having a limited amount of labeled data compared to the total data collected. To address this challenge, applying semi-supervised methods becomes crucial. Our ongoing efforts are focused on devising a semi-supervised variant of GP-VIB. This adaptation involves integrating a generative model for unlabeled instances to approximate an accurate posterior for the data. By incorporating semi-supervised techniques, we aim to enhance the model's ability to learn from both labeled and unlabeled data, potentially improving its performance in prevalent real-world scenarios that are characterized by limited labeled instances.

## References

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. 2016.

[2] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[3] Robert Bamler and Stephan Mandt. Structured black box variational inference for latent time series models. *arXiv preprint arXiv:1707.01069*, 2017.

[4] Alessio Benavoli, Giorgio Corani, and Francesca Mangili. Should we really use post-hoc tests based on mean-ranks? *The Journal of Machine Learning Research*, 17(1):152–161, 2016.

[5] Nestor Cabello, Elham Naghizade, Jianzhong Qi, and Lars Kulik. Fast, accurate and interpretable time series classification through randomization. *arXiv preprint arXiv:2105.14876*, 2021.

[6] Ranak Roy Chowdhury, Xiyuan Zhang, Jingbo Shang, Rajesh K Gupta, and Dezhi Hong. Tarnet: Task-aware reconstruction for time-series transformer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 212–220, 2022.

[7] Ranak Roy Chowdhury, Xiyuan Zhang, Jingbo Shang, Rajesh K Gupta, and Dezhi Hong. Tarnet: Task-aware reconstruction for time-series transformer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 212–220, 2022.

[8] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, 307:72–77, 2018.

[9] Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.

[10] Angus Dempster, Daniel F Schmidt, and Geoffrey I Webb. Hydra: Competing convolutional kernels for fast and accurate time series classification. *Data Mining and Knowledge Discovery*, pages 1–27, 2023.

[11] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

[12] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*, 2021.

[13] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*, pages 1651–1661. PMLR, 2020.

[14] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

[15] Ge Gao, Qitong Gao, Xi Yang, Miroslav Pajic, and Min Chi. A reinforcement learning-informed pattern mining framework for multivariate time series classification. In *In the Proceeding of 31th International Joint Conference on Artificial Intelligence (IJCAI-22)*, 2022.

[16] Salvador Garcia and Francisco Herrera. An extension on" statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of machine learning research*, 9(12), 2008.

[17] Antoine Guillaume, Christel Vrain, and Wael Elloumi. Random dilated shapelet transform: A new approach for time series shapelets. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 653–664. Springer, 2022.

[18] Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F Schmidt, Jonathan Weber, Geoffrey I Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. Inceptiontime: Finding alexnet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020.

[19] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate lstm-fcns for time series classification. *Neural networks*, 116:237–245, 2019.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *stat*, 1050:1, 2014.

[21] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv e-prints*, pages arXiv–1511, 2015.

[22] Guozhong Li, Byron Choi, Jianliang Xu, Sourav S Bhowmick, Kwok-Pan Chun, and Grace Lai-Hung Wong. Shapenet: A shapelet-neural network approach for multivariate time series classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8375–8383, 2021.

[23] Benjamin Lucas, Ahmed Shifaz, Charlotte Pelletier, Lachlan O'Neill, Nayyar Zaidi, Bart Goethals, François Petitjean, and Geoffrey I Webb. Proximity forest: an effective and scalable distance-based classifier for time series. *Data Mining and Knowledge Discovery*, 33(3):607–635, 2019.

[24] Matthew Middlehurst and Anthony Bagnall. The freshprince: A simple transformation based pipeline time series classifier. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 150–161. Springer, 2022.

[25] Matthew Middlehurst, James Large, Gavin Cawley, and Anthony Bagnall. The temporal dictionary ensemble (tde) classifier for time series classification. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, pages 660–676. Springer, 2021.

[26] Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron Bostrom, and Anthony Bagnall. Hive-cote 2.0: a new meta ensemble for time series classification. *Machine Learning*, 110(11-12):3211–3243, 2021.

[27] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *arXiv preprint arXiv:2304.13029*, 2023.

[28] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

[29] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(3):1–31, 2013.

[30] Patrick Schäfer and Ulf Leser. Weasel 2.0–a random dilated dictionary transform for fast, accurate and memory constrained time series classification. *arXiv preprint arXiv:2301.10194*, 2023.

[31] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[32] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[33] Chang Wei Tan, Angus Dempster, Christoph Bergmeir, and Geoffrey I Webb. Multirocket: multiple pooling operators and transformations for fast and effective time series classification. *Data Mining and Knowledge Discovery*, 36(5):1623–1646, 2022.

[34] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[35] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

[36] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

[37] Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6845–6852, 2020.