Considering Nonstationary within Multivariate Time Series with Variational Hierarchical Transformer for Forecasting

Muyao Wang, Wenchao Chen *, Bo Chen

National Laboratory of Radar Signal Processing Xidian University, Xi'an, Shaanxi, China
muyaowang@stu.xidian.edu.cn, chenwenchao@xidian.edu.cn, bchen@mail.xidian.edu.cn

Abstract

The forecasting of **M**ultivariate **T**ime **S**eries (MTS) has long been an important but challenging task. Due to the non-stationary problem across long-distance time steps, previous studies primarily adopt stationarization method to attenuate the non-stationary problem of the original series for better predictability. However, existing methods always adopt the stationarized series, which ignores the inherent non-stationarity, and has difficulty in modeling MTS with complex distributions due to the lack of stochasticity. To tackle these problems, we first develop a powerful hierarchical probabilistic generative module to consider the non-stationarity and stochastity characteristics within MTS, and then combine it with transformer for a well-defined variational generative dynamic model named **H**ierarchical **T**ime series **V**ariational **Trans**former (HTV-Trans), which recovers the intrinsic non-stationary information into temporal dependencies. Being a powerful probabilistic model, HTV-Trans is utilized to learn expressive representations of MTS and applied to forecasting tasks. Extensive experiments on diverse datasets show the efficiency of HTV-Trans on MTS forecasting tasks.

Introduction

Multivariate time series (MTS) is an important type of data that arises from a wide variety of domains, including internet services, industrial devices, health careand finance, to name a few. However, the forecasting of MTS has always been a challenging problem as there exists not only complex temporal dependencies, as shown in the red box in Fig. 1, but also inherently stochastic components, as shown in the green box in Fig. 1. Moreover, there exist non-stationary issues, as shown in the blue box in Fig. 1, which haves a huge impact on predictive performance. To model the temporal dependencies of MTS, many dynamic methods based on recurrent neural networks (RNNs) have been developed (Malhotra et al. 2016; Zhang et al. 2019; Bai et al. 2019; Tang et al. 2020; Yao et al. 2018). Meanwhile, to consider the stochasticity, some probabilistic dynamic methods have also been developed (Dai et al. 2021, 2022; Chen et al. 2020, 2022; Salinas et al. 2020). With the development of Transformer (Vaswani et al. 2017) and due to its ability to capture long-range dependencies (Wen et al. 2022; Dosovitskiy et al. 2021; Chen et al. 2021), and interactions, which is especially attractive for time series forecasting, there is a recent trend to construct Transformer based MTS forecasting methods and have achieved promising results in learning expressive representations for MTS forecasting tasks. Recently, there are lots of efficient Transformer-based forecasting methods, such as Autoformer (Wu et al. 2021), FEDformer (Zhou et al. 2022), Pyraformer (Liu et al. 2021), Crossformer (Zhang and Yan 2023) and so on. Despite the advanced architectural design, it is still hard for Transformers to predict real-world time series because of the non-stationarity of data. Non-stationary time series is characterized by the continuous change of statistical properties and joint distribution over time, making them less predictable (Hyndman and Athanasopoulos 2018). In previous works, they usually pre-process the time series by stationarization (Ogasawara et al. 2010a; Passalis et al. 2019a; Kim et al. 2021), which can attenuate the non-stationarity of raw time series and provide more stable data distribution for deep models. However, these MTS stationarization methods ignore the non-stationarity of real-world series, which will result in the over-stationarization problem (Liu et al. 2022).

Non-stationary transformer (Liu et al. 2022) is proposed to tackle the over-stationarization problem in the Transformers by approximating distinguishable attentions learned from raw series, which is a method limited to a specific transformer situation. Moreover, because of the deterministic architectural designs, the methods mentioned before still face challenges when it comes to predicting real-world time series due to the non-deterministic data caused by noise data. Therefore, in this paper, we analyze the series forecasting task from another point of view to bring the non-deterministic and non-stationarity back to the deep models, especially for the transformer, and propose a novel dynamic hierarchical generative module to effectively capture both inherent properties.

Moving beyond the constraints of previous work, considering the non-determinism and non-stationarity within MTS and enhancing the representative power of deep models, we develop a **H**ierarchical **T**ime series **V**ariational **Trans**former (HTV-Trans), which is a well-defined probabilistic dynamic model obtained by combining a proposed **H**ierarchical **T**ime series **P**robabilistic **G**enerative **M**odule (HTPGM), as illustrated in Fig. 2 (a)(b), with the Transformer block, as in

*Correspondence to Wenchao Chen (chenwenchao@xidian.edu.cn)

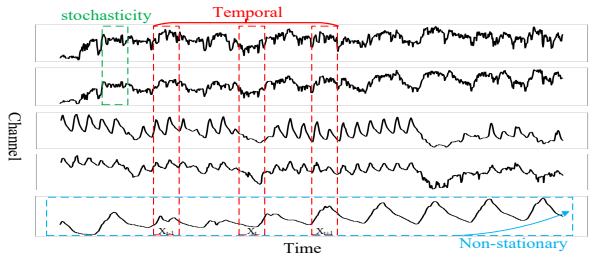


Figure 1: The temporal dependency, stochasticity and non-stationarity within MTS.

Fig. 2 (c). Specifically, HTPGM module is able to get the multi-scale non-deterministic and non-stationary representations of the original MTS, which can be served to guide the forecasting tasks with transformer block. In addition, we introduce an autoencoding variational inference scheme for efficient inference and a joint optimization objective that combines forecasting and reconstruction loss to recover the non-deterministic and non-stationary time-series representation into Transformer. The main contributions of our work are summarized as follows:

- For MTS forecasting, we propose HTPGM module, which is able to consider the non-stationary information within the MTS with a hierarchical multi-scale generative structure, thus avoiding the over-stationarization problem.
- We develop HTV-Trans, a probabilistic dynamic model with HTPGM as a generative module, which can consider the non-deterministic and non-stationary issues within the temporal dependencies of MTS.
- For optimization, we introduce an autoencoding inference scheme with a combined prediction and reconstruction loss to enhance the representation power of MTS.
- Experiments on different datasets illustrate the efficiency of our model on MTS forecasting task.

Backgrounds

Multivariate Time Series Forecasting

In recent years, transformer models have subsequently emerged and have shown great power in sequence modeling. To overcome the quadratic computational growth in relation to sequence length, subsequent works have aimed to reduce the complexity of Self-Attention. In particular, for time series forecasting, Informer(Zhou et al. 2021) extends Self-Attention with a KL-divergence criterion to select dominant queries. Reformer(Kitaev, Kaiser, and Levskaya 2019) introduces local-sensitive hashing (LSH) to approximate attention by allocating similar queries. Not only have these models been improved by reduced complexity, they have also developed more complex building blocks for time series forecasting. Autoformer(Wu et al. 2021) fuses decomposition blocks into a canonical structure and develops Auto-Correlation to discover series-wise connections. Pyraformer (Liu et al. 2021) designs a pyramid attention module (PAM) to capture temporal dependencies at different hierarchies. Other deep Transformer models have also achieved remarkable performance. Fedformer(Zhou et al. 2022) designs two attention modules, which use Fourier transform and wavelet transform to process attention operations in the frequency domain. Non-stationary transformer(Liu et al. 2022) designs a de-stationary attention, which approximates the distinguishable attention in non-stationary sequences to solve the problem of excessive stationarity. Crossformer(Zhang and Yan 2023) proposes a Two-Stage-Attention (TSA) layer to capture the cross-time and crossdimension dependency of the embedded array and show its effectiveness over previous state-of-the-arts. In this paper, we take a different approach from previous works that focus on transformer architectural design. In this paper, we propose a novel approach to address the non-stationary and non-deterministic properties of time series, which are essential characteristics of this type of data (Kim et al. 2021; Liu et al. 2022).

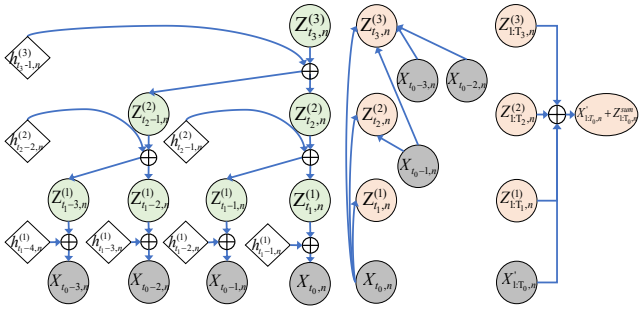
Non-stationary Problems of Time Series

While stationarity is important to the predictability of time series (Kim et al. 2021; Liu et al. 2022), real-world series always present non-stationarity. To tackle this problem, the classical statistical method ARIMA (Box 1976) stationarizes the time series through differencing. As for deep models, since the distribution-varying problem accompanied by non-stationarity makes deep forecasting even more intractable, stationarization methods are widely explored and always adopted as the pre-processing for deep model inputs. Adaptive Norm (Ogasawara et al. 2010b) applies z-score normalization for each series fragment by global statistics of a sampled set. DAIN (Passalis et al. 2019b) employs a nonlinear neural network to adaptively stationarize time series with observed training distribution. RevIN (Kim et al. 2021) introduces a two-stage instance normalization that transforms model input and output respectively to reduce the discrepancy of each series, which brings great benefit to the model's capability of modeling non-stationary time series. However, Non-stationary transformer (Liu et al. 2022) find out that directly stationarizing time series will damage the model's capability of modeling specific temporal dependency, which is named over-stationarization and tackle the problem by approximating distinguishable attentions learned from the original series. Given this over-stationarization issue, unlike Non-stationary-transformer method, HTV-Trans further develops a hierarchical generative module to capture the multi-scale statistical characteristics of the original input time series, which can bring more intrinsic non-stationarity of the original series back to latent representation for time series forecasting.

Method

Problem Definition

Defining the n-th MTS as $\boldsymbol{x}_n = \{\boldsymbol{x}_{1,n}, \boldsymbol{x}_{2,n}, ..., \boldsymbol{x}_{T,n}\}$, where $n = 1, ...N$ and N is the number of MTS. T is the duration of $\boldsymbol{x}_n$ and the observation at time t, $\boldsymbol{x}_{t,n} \in \mathbb{R}^V$, is a V dimensional vector where V denotes the number of channels, thus $\boldsymbol{x}_n \in \mathbb{R}^{T \times V}$.



a) Generation b) Inference c) Fusion

Figure 2: Graphical illustration of different operations of the HTV-Trans: (a) generative process of HTPGM, (b) the inference scheme of HTPGM. (c) the fusion of different scale information and stationarization input series for forecasting.

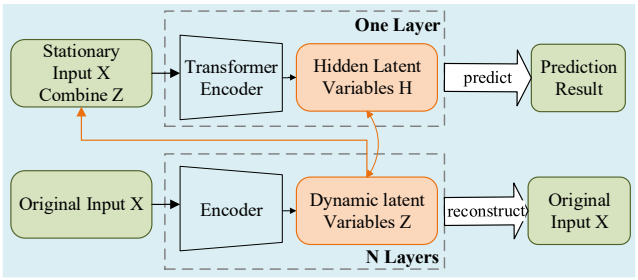


Figure 3: the whole framework of HTV-Trans.

Time Series Stationarization

Non-stationary time series present a significant challenge for deep models in forecasting tasks, as they tend to struggle with generalization to series exhibiting variations in mean and standard deviation during forecasting. This is due to the inherent difficulty of modeling time series with changing statistical properties. The pilot work, RevIN (Kim et al. 2021) and Non-stationary transformer (Liu et al. 2022) both put forward the instance normalization to each input and restores the statistics to the corresponding output in a similar way, which makes each series follow a similar distribution. Empirical evidence has shown that this design is effective, but Non-stationary transformer introduces an alternative method called Series Stationarization that requires fewer computational resources. Based on these considerations, we adopt Series Stationarization for normalizing our input data.

Normalize module: To attenuate the non-stationarity of each input series, Series Stationarization conducts normalization on the temporal dimension by a sliding window over time. Here are the equations.

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{x}} &= \tfrac{1}{T}\textstyle\sum_{i=1}^{T} \boldsymbol{x}_i, \boldsymbol{\sigma}_{\mathbf{x}}^2 = \tfrac{1}{T}\textstyle\sum_{i=1}^{T} (\boldsymbol{x}_i - \boldsymbol{\mu}_{\mathbf{x}})^2 \\ \boldsymbol{x}_i' &= \tfrac{1}{\boldsymbol{\sigma}_{\mathbf{x}}} \odot (\boldsymbol{x}_i - \boldsymbol{\mu}_{\mathbf{x}}) \end{aligned} \tag{1}$$

where $\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\sigma}_{\mathbf{x}} \in \mathbb{R}^{V\times 1}$, $\frac{1}{\boldsymbol{\sigma}_{\mathbf{x}}}$ means the element-wise division and $\odot$ is the element-wise product. It is obvious that the Series Stationarization aims to reduce the distributional differences among individual input time series, thereby stabilizing the distribution of model inputs.

Denormalize module: After the base model $\boldsymbol{H}$ predicting the future value with length-$\boldsymbol{T}$, we adopt De-normalization to transform the model output $\boldsymbol{y}' = [\boldsymbol{y}_1', \boldsymbol{y}_2', ..., \boldsymbol{y}_t'] \in \mathbb{R}^{T\times V}$ with $\boldsymbol{\sigma}_x$ and μ_x , thus obtain $\boldsymbol{y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_t]$ as the eventual forecasting results. The De-normalization module can be formulated as follows:

$$\boldsymbol{y}' = \boldsymbol{H}\ (\boldsymbol{x}'), \boldsymbol{y}_i = \boldsymbol{\sigma}_{\boldsymbol{x}} \odot (\boldsymbol{y}_i' + \boldsymbol{\mu}_{\mathbf{x}}) \tag{2}$$

Because of the two-stage transformation, the base models will receive stationarized inputs, which follow a stable distribution and are easier to generalize. These designs also make the model equivariant to translational and scaling perturbance of time series, which benefits real-world series forecasting.

Hierarchical Time Series Variational Transformer

Analysis of over-stationarization: Although the statistics of each output time series are explicitly restored to the corresponding original distribution of input series by De-normalization, the non-stationarity of the original series cannot be fully restored only in this way. According to the conclusion from the Non-stationary transformer(Liu et al. 2022), the undermined effects caused by over-stationarization happen inside the deep model, especially in the calculation of attention. Furthermore, non-stationary time series are fragmented and normalized into several series chunks with the same mean and variance, which follow more similar distributions than the raw data before stationarization. Thus, the model is more likely to generate over-stationary and uneventful outputs, which is irreconcilable with the natural non-stationarity of the original series. To address the over-stationarization issue caused by Series Stationarization, we propose a novel Hierarchical Time series Variational Autoencoder(HTPGM) that can solve this problem more completely. Similar to the Non-stationary transformer, we fuse the normalized data and the non-stationary information (as shown in Fig. 2 (c)) provided by the hierarchical distribution to solve the problem of similar attention caused by the similar distribution of the input data. Furthermore, our hierarchical generative module can provide fine-grained data distribution information through the different hidden layers distribution at different scales, which can boost the non-stationary series predictive capability of the base model.

Hierarchical generative module: As previously mentioned, to tackle the over-stationarization problem caused by Series Stationarization, Non-stationary transformer proposes a novel De-stationary Attention mechanism, which focuses on the attention caused by stationarization and discovers the particular temporal dependencies from original non-stationary data. From another point of view, we introduce a hierarchical probabilistic generative module to consider the original series statistical information through the generative process in our proposed model. The whole generative module is illustrated in Fig. 2 (a). This module aims to consider the inherent uncertainty and intrinsic non-stationarity of the original input time series and enhance the capacity of the model to handle

complex, non-deterministic, and non-stationary time series distributions. Our model mainly starts with three aspects to address the issues which are mentioned previously.

Vanished non-stationary information supplement: Firstly, given the intrinsic non-stationarity of input series, we consider that the post distribution of HTPGM should depend on the original input series as shown in Fig. 2. Due to the reconstruction of raw input series, the intrinsic non-stationarity can be learnable through the generative process.

Multi-scale non-stationary information: Secondly, we design a generative process in which hidden layer distribution has different scales for HTPGM. The time dimension of hidden variables $\boldsymbol{z}_i$ from the bottom layer to the top layer is gradually reduced. Our design aims to let the distribution of different scales capture more fine-grained original time series information. Because the way of information fusion is achieved by using the nearest neighbor interpolation to unify the dimension of hidden layer distribution, this design is equivalent to setting several windows to capture the distribution in different small chunks of raw time series. Due to the different scales of hidden layer distribution, the hierarchical generative process can provide fine-grained distribution information, which is stored in $\boldsymbol{z}_{sum}$ as shown in Fig. 2 (c).

Robust representation: Our proposed generative module is also capable of solving the non-deterministic problems of time series by utilizing the probabilistic component. The probabilistic component enables the incorporation of non-determinism into the representation of time series, by introducing a degree of randomness and uncertainty in the generative process. This characteristic of the probabilistic generative module makes it robust to the noise in the input, allowing for the modeling of non-deterministic time series. Overall, our generative module can extract a non-stationary and roust representation of time series.

Generation: We propose a special way that incorporates the output of transformer $\boldsymbol{h_t}$(as shown in Fig. 2 (a)) into the generative process of HTPGM. The generative process is illustrated in Fig. 2 (a). This operation enhances the ability of $\boldsymbol{h_t}$ and latent variable $\boldsymbol{z_i}$ to reconstruct the original input series, resulting in an expressive and underlying non-stationary representation of the time series. The detail of the generative process is defined as

$$
\begin{aligned}
&\boldsymbol{z}^L_{t_L,n} \sim \mathcal{N}\left(\boldsymbol{W}^L_{t_L,n}\boldsymbol{h}^L_{t_L,n}, \boldsymbol{I}\right), \quad \cdots\cdots \\
&\boldsymbol{z}^i_{t_i,n}, ..., \boldsymbol{z}^i_{t_i+K_i,n} \sim \mathcal{N}\left(\boldsymbol{\mu}^{i+1}_{t_{i+1},n}\left(\boldsymbol{z}^{i+1}_{t_{i+1},n}\right), \text{diag}\left(\boldsymbol{\sigma}^i\right)\right) \\
&\boldsymbol{\mu}^{i+1}_{t_{i+1},n}\left(\boldsymbol{z}t_{i+1}, n^{i+1}\right) = f(\boldsymbol{W}^{i+1}_{\boldsymbol{z},\boldsymbol{\mu}}\boldsymbol{z}^{i+1}_{t-1,n} + \boldsymbol{W}^i_{h\mu}\boldsymbol{h}^{i+1}_{t_{i+1}-1,n}) \\
&\cdots\cdots, \quad \boldsymbol{x}_{t_0,n} \sim \mathcal{N}\left(\boldsymbol{\mu}^1_{t,n}, \text{diag}\left(\boldsymbol{\sigma}^x\right)\right), \\
&\boldsymbol{\mu}^1_{t,n} = f\left(\boldsymbol{W}^{i+1}_{z,\mu} z^{i+1}_{t-1,n} + \boldsymbol{W}^1_{h\mu}\boldsymbol{h}^{i+1}_{t_{i+1}-1,n}\right)
\end{aligned}
$$

where $\boldsymbol{W}^i_{\boldsymbol{z}\mu}, \boldsymbol{W}^i_{\boldsymbol{h}\mu} \in \mathbb{R}^{K^i \times V}$, are all learnable parameters of the generate network, K^i, t_i are changed with different layer i. $f(\cdot)$ refers to the non-linear activation function. $\boldsymbol{\mu}^i_{t,n}$ and $\boldsymbol{\sigma}^i_{t,n}$ are means and covariance parameters of $\boldsymbol{z}^i_{t,n}$. $\boldsymbol{h}_{t-1,n} \in \mathbb{R}^{T \times V}$ denotes the deterministic latent states of our forecasting module. We combine $\boldsymbol{z}^i_{t,n}$ and $\boldsymbol{h}_{t-1,n}$ into the generative process to consider the temporal dependencies and the stochasticity. The prior information is passed down through the generative process from the top layer to the bottom layer of the HTPGM. This hierarchical generative process allows each layer of HTPGM to learn the non-stationary information from different perspectives. It enables the model to capture a diverse range of temporal dependencies and variations within the original input time series, resulting in a more robust and non-stationary multi-scale representation of the original series. This hierarchical approach allows the model to effectively handle complex, non-stationary time series, leading to improved performance in forecasting tasks.

Hierarchical variational transformer: In order to maximize the advantages of our hierarchical generative module, we have designed a transformer-based prediction model, which we call HTV-trans (as shown in Fig. 3). In this prediction model, we use a one-layer transformer encoder as our model basic feature extractor and a MLP as a forecasting block to output the prediction in one step.

Transformer block: Given the long time series dependency, we choose a transformer encoder to capture the dynamic information from the normalized input series. As mentioned previously, we consider that the $\boldsymbol{z}_{sum}$ as shown in Fig. 2 (c) can store the vanished non-stationary information of original time series. Thus, we decide to integrate the normalized series and $\boldsymbol{z}_{sum}$ as input to solve the over stationarizaiton problem. Specifically, we introduce a **M**ulti-head **S**elf **A**ttention (MSA) block to capture the temporal dependence as

$$
\begin{aligned}
\boldsymbol{O} &= \text{MSA}(\text{Embedding}(\boldsymbol{X}^{'}) + \\
&\quad \alpha\text{sum}(\text{Interpolate}(\boldsymbol{Z}_i))) = \text{con}\left(\boldsymbol{H}_1, \ldots, \boldsymbol{H}_m\right) \\
\boldsymbol{H}_i &= \text{SA}\left(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i\right) = \text{softmax}(\tfrac{\boldsymbol{Q}_i^T \boldsymbol{K}_i}{\sqrt{d_K}})\boldsymbol{V}_i
\end{aligned} \tag{3}
$$

where $\boldsymbol{X}^{'} \in \mathbb{R}^{V \times T}$, denotes the normalized input, Embedding$(\cdot)$ denotes the time feature embedding which is mentioned in (Zhou et al. 2021), sum(Z) denotes the sum of the hidden layers of HTPGM and con$(\cdot)$ means concatenate operation, $\boldsymbol{Q}_i = \boldsymbol{W}^i_Q(\boldsymbol{X}^{'} + \text{sum}(\text{Interpolate}(\boldsymbol{Z}))) \in \mathbb{R}^{d_k \times T}$, $\boldsymbol{K}_i = \boldsymbol{W}^i_K(\boldsymbol{X}^{'} + \text{sum}(\text{Interpolate}(\boldsymbol{Z}))) \in \mathbb{R}^{d_k \times T}$, where $i \in \{1, 2, ..., m\}$ and m is the number of heads. $\boldsymbol{O} \in \mathbb{R}^{(V \times T \times m)}$ is the output of MSA block. It is important to note that we introduce a new parameter α, which decides the balance between the stationary information and non-stationary information. To prove the validity of α, we introduce an ablation study of α in section 5. After combining temporal, non-stationary and non-deterministic information of MTS into $\tilde{\boldsymbol{h}} \in \mathbb{R}^{V \times T}$, feed forward network blocks are further applied for exploring expressive representations of MTS and getting the dynamic latent states $\boldsymbol{h} \in \mathbb{R}^{V \times T}$.

MLP for forecasting: Different from other transformer models, we select the MLP decoder for forecasting tasks due to the fact that the traditional transformer decoder tends to fit time series data too rapidly, leading to inadequate learning of effective representations in the latent layers of HTPGM. After a number of experiments, we find that the MLP decoder is more appropriate for our generative module. In order to output the prediction in one step, the MLP output size is set to be the same as the prediction length. Combining HTPGM and Transformer, we finally develop HTV-Trans, a

novel hierarchical probabilistic generative dynamic model. The graphical illustration of the whole framework is shown in Fig. 3.

Multi-scale inference network: Following VAE-based models, we define a Gaussian distributed variational distribution $q(\boldsymbol{z}_{t,n}) = \mathcal{N}(\boldsymbol{\mu}_{t,n}, \text{diag}(\boldsymbol{\sigma}_{t,n}))$ to approximate the true posterior distribution $p(\boldsymbol{z}_{t,n}|-)$, and map the dynamic input series $\hat{\boldsymbol{x}}_{t,n}$ to their parameters as:

$$\begin{aligned} q(\boldsymbol{z}_{t,n}) &= \mathcal{N}(\boldsymbol{\mu}_{t,n}, \text{diag}(\boldsymbol{\sigma}_{t,n})) \\ \boldsymbol{\mu}^i_{t,n}(\boldsymbol{x}) &= f\left(\boldsymbol{C}^i_{x\mu}(\boldsymbol{x}) + \boldsymbol{b}^i_{x\mu}\right) \\ \boldsymbol{\sigma}^i_{t,n}(\boldsymbol{x}) &= \text{Softplus}\left(f\left(\boldsymbol{C}^i_{x\sigma}(\boldsymbol{x}) + \boldsymbol{b}^i_{x\sigma}\right)\right) \end{aligned} \tag{4}$$

where $\boldsymbol{C}^i_{x\mu}, \boldsymbol{C}^i_{x\sigma} \in \mathbb{R}^{K \times V'}$, $\boldsymbol{b}^i_{x\mu}, \boldsymbol{b}^i_{x\sigma} \in \mathbb{R}^K$ are all learnable parameters of the inference network. $f(\cdot)$ refers to the non-linear activation function. Based on the structure of the inference network as shown in Fig. 3 (b), the posterior of probabilistic latent variables of HTPGM are approximated by multi-scale original input series, thus enabling richer latent representations for HTPGM.

Model Training

As mentioned in (Cao et al. 2020; Wen et al. 2022), the prediction-based model is expert in capturing the periodic information of the MTS, while the reconstruction-based model tends to explore the global distribution of the MTS. To combine the complementary advantages of them for facilitating the representation capability of MTS, we formulate the optimization function as the combination of both prediction and reconstruction losses and define the marginal likelihood as

$$\begin{aligned} &P\left(\mathcal{D} \mid \{\boldsymbol{W}^{(l)}\}_{l=1}^L\right) = \int \left[\prod_{t=1}^T p\left(\boldsymbol{x}_{t,n} \mid \boldsymbol{z}^{(1)}_{t,n}\right)\right. \\ &+ \left[\prod_{l=1}^L \prod_{t=1}^T p\left(\boldsymbol{z}^{(l)}_{t,n} \mid \boldsymbol{z}^{(l+1)}_{t,n}\right)\right] \\ &\left.+ \left[\prod_{l=1}^L p\left(\boldsymbol{x}_{T,n} \mid \boldsymbol{h}^{(l)}_{1:T-1,n}\right)\right] d\boldsymbol{z}^{1:L}_{1:T,n}\right. \end{aligned} \tag{5}$$

where the first and the second terms are reconstruction and prediction loss separately. Similar to VAEs, with the inference network and variational distribution in Eq. (4), the optimization objective of HTV-Trans can be achieved by maximizing the evidence lower bound (ELBO) of the log marginal likelihood, which can be computed as

$$\begin{aligned} &\mathcal{L} = \sum_{n=1}^N \left[\sum_{t=1}^T \mathbb{E}_{q\left(\boldsymbol{z}^{(1)}_{t,n}\right)} \left[\ln p\left(\boldsymbol{x}_{t,n} \mid \boldsymbol{z}^{(1)}_{t,n}\right)\right]\right. \\ &+\gamma \mathbb{E}_{q\left(\boldsymbol{z}^{(1)}_{t,n}\right)} \left[\ln p\left(\boldsymbol{x}_{T,n} \mid h^{(1)}_{1:T-1,n}\right)\right] \\ &\left.- \sum_{t=1}^T \sum_{l=1}^L \mathbb{E}_{q(\boldsymbol{z}^l_{t,n})} \left[\ln \frac{q\left(\boldsymbol{z}^{(l)}_{t,n} | \boldsymbol{x}_{t,n}\right)}{p\left(\boldsymbol{z}^{(l)}_{t,n} | \boldsymbol{z}^{(l+1)}_{t,n}\right)}\right]\right] \end{aligned} \tag{6}$$

where $\gamma > 0$ is a hyper-parameter to balance the prediction and the reconstruction losses, which is chosen by grid search on the validation set. The detailed procedures of the optimization of HTV-Trans are summarized in Appendix.

Experiments

We conduct extensive experiments to prove the effectiveness of our proposed model. Code is available at https://github.com/flare200020/HTV_Trans.

Datasets and Set Up

We evaluate the effectiveness of our model on seven datasets for forecasting, including ETTh1, ETTh2, ETTm1, ETTm2, Illness, Weather and Exhcange-rate (Liu et al. 2022). The results are either quoted from the original papers or reproduced with the code provided by the authors. The way of data preprocessing is the same as (Liu et al. 2022). The summary statistics of these datasets and other implementation details are described in Appendix.

Forecasting Main Results

We deploy two widely used metrics, Mean Absolute Error (MAE) and Mean Square Error (MSE) (Zhou et al. 2021) to measure the performance of forecasting models. Six popular state of the art methods are compared here, including Crossformer (Zhang and Yan 2023), Non-stationary transformer (Liu et al. 2022), Fedformer (Zhou et al. 2022); Pyraformer (Liu et al. 2021); Informer (Zhou et al. 2021) and Autoformer (Wu et al. 2021). We note that the experiment settings used here are the same as Non-stationary transformer mentioned (Liu et al. 2022). Table 1 presents the overall prediction performance, which is the average MAE and MSE on five independent runs, and the best results are highlighted in boldface. Evaluation results demonstrate that our proposed method outperforms other Transformer-based approaches in most settings, particularly in long-term forecasting tasks. This suggests the effectiveness of our model structure in modeling complex and long-range temporal dependencies. Our method achieves superior performance on almost all datasets, suggesting its ability to capture non-deterministic and non-stationary complex temporal dependencies in MTS, leading to expressive representations and promising prediction outcomes.

Ablation Study

Prediction quality evaluation: To investigate the contributions of each module in our proposed framework, wo conduct the ablation study by comparing the prediction results of Transformer, Transformer with traditional **H**ierarchical VAE (HVAE), and our HTV-Transformer on the ETTm2 dataset and visualize the results in Figure 4. We find that HTPGM enhances the non-stationary and non-deterministic forecasting ability of Transformers from different angles significantly. Obviously, as demonstrated in Figure 4, traditional Transformer tends to produce output series with high stationarity and volatility, neglecting the non-deterministic and non-stationary nature of real-world data. When the transformer combines the traditional HVAE, the prediction tends to be smooth, but it still fails to capture the non-stationarity of real-world time series due to its poor ability to reconstruct the original series. Unlike the traditional HVAE, the dynamic variable h is introduced in the generative process in our model to help the hidden layer i better learn the distribution of input time series. Besides that, the incorporation of dynamic prior

Models		HTV-Trans		Autoformer		Informer		NS Transformer		Fedformer		Pyrafomer		Crossformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.389	**0.396**	0.536	0.548	0.984	0.786	0.513	0.491	**0.376**	0.419	0.783	0.657	0.431	0.441
	192	0.445	**0.422**	0.543	0.551	1.027	0.791	0.534	0.504	0.420	0.448	0.863	0.709	**0.411**	0.440
	336	0.487	**0.440**	0.615	0.592	1.032	0.774	0.588	0.535	0.459	0.465	0.941	0.753	**0.441**	0.461
	720	**0.489**	**0.455**	0.599	0.600	1.169	0.858	0.643	0.616	0.506	0.507	1.042	0.819	0.515	0.518
ETTh2	96	**0.300**	**0.338**	0.492	0.517	2.826	1.330	0.476	0.458	0.346	0.388	1.380	0.943	0.860	0.691
	192	**0.382**	**0.391**	0.556	0.551	6.186	2.070	0.512	0.493	0.429	0.439	3.809	1.634	1.026	0.729
	336	**0.377**	**0.405**	0.572	0.578	5.268	1.942	0.552	0.551	0.482	0.480	4.282	1.792	1.110	0.768
	720	**0.412**	**0.438**	0.580	0.588	3.667	1.616	0.562	0.560	0.463	0.474	4.252	1.790	2.151	1.134
ETTm1	96	0.337	**0.354**	0.523	0.488	0.615	0.556	0.386	0.398	0.378	0.418	0.536	0.506	**0.320**	0.373
	192	**0.366**	**0.374**	0.543	0.498	0.723	0.620	0.459	0.444	0.426	0.441	0.539	0.520	0.407	0.437
	336	**0.412**	**0.396**	0.675	0.551	1.300	0.908	0.495	0.464	0.445	0.459	0.720	0.635	0.417	0.433
	720	**0.484**	**0.434**	0.720	0.573	0.972	0.744	0.585	0.516	0.543	0.490	0.940	0.740	0.610	0.554
ETTm2	96	**0.178**	**0.259**	0.255	0.339	0.365	0.453	0.192	0.274	0.203	0.287	0.409	0.488	0.490	0.487
	192	**0.248**	**0.301**	0.281	0.340	0.533	0.563	0.280	0.339	0.269	0.328	0.673	0.641	0.922	0.711
	336	**0.311**	**0.341**	0.339	0.372	1.363	0.887	0.334	0.361	0.325	0.366	1.210	0.846	0.770	0.590
	720	**0.405**	**0.394**	0.422	0.419	3.379	1.388	0.417	0.413	0.421	0.415	4.044	1.526	0.920	0.730
Weather	96	0.181	**0.223**	0.266	0.336	0.300	0.384	0.173	0.223	0.217	0.296	0.354	0.392	**0.163**	0.226
	192	**0.216**	**0.257**	0.307	0.367	0.597	0.598	0.245	0.285	0.276	0.336	0.673	0.597	0.219	0.278
	336	**0.278**	**0.293**	0.359	0.395	0.578	0.523	0.321	0.338	0.339	0.380	0.634	0.592	0.280	0.322
	720	**0.341**	**0.344**	0.419	0.428	1.059	0.741	0.414	0.410	0.403	0.428	0.942	0.723	0.360	0.388
ILI	24	**2.113**	**0.899**	3.483	1.287	5.764	1.677	2.294	0.945	3.228	1.660	5.800	1.693	3.041	1.186
	36	1.833	**0.800**	3.103	1.148	4.755	1.467	**1.825**	0.848	2.679	1.080	6.043	1.733	3.406	1.232
	48	**2.012**	**0.862**	2.669	1.085	4.763	1.469	2.010	0.900	2.622	1.078	6.213	1.763	3.459	1.221
	60	**2.114**	**0.896**	2.770	1.125	5.264	1.564	2.178	0.963	2.857	1.157	6.531	1.814	3.640	1.305
Exchange	96	**0.086**	**0.205**	0.197	0.323	0.847	0.752	0.111	0.237	0.148	0.271	0.852	0.780	0.246	0.392
	192	**0.190**	0.304	0.300	0.369	1.204	0.895	0.219	0.335	0.271	**0.280**	0.993	0.858	0.889	0.720
	336	**0.370**	**0.431**	0.509	0.524	1.672	1.036	0.421	0.476	0.460	0.500	1.240	0.958	1.375	0.935
	720	**0.901**	**0.710**	1.447	0.941	2.478	1.310	1.092	0.769	1.195	0.841	1.711	1.093	1.978	1.175

Table 1: The input sequence length is set to 36 for ILI and 96 for the others. Multivariate results with predicted length as {96, 192, 336, 720} on the six datasets and {24, 36, 48, 60} on the ILI dataset, lower scores are better. Metrics are averaged over 5 runs, best results are highlighted in bold.

Architecture	Optimization Objective	Multivariate time series forecasting								
		ETTH2			Weather			Exchange		
		96	192	336	96	192	336	96	192	336
Transformer	Prediction	0.368	0.418	0.439	0.230	0.286	0.326	0.246	0.320	0.461
Transformer with traditional HVAE	Prediction	0.367	0.417	0.433	0.232	0.275	0.325	0.249	0.324	0.479
	Combine	0.362	0.416	0.433	0.228	0.270	0.325	0.241	0.319	0.473
Transformer with our HTPGM	Prediction	0.367	0.414	0.437	0.230	0.272	0.334	0.249	0.322	0.486
	Combine	**0.338**	**0.391**	**0.405**	**0.223**	**0.257**	**0.293**	**0.205**	**0.304**	**0.431**

Table 2: Ablation study of HTV-Trans on forecasting tasks. (Transformer denotes the transformer with MLP decoder)

information focuses on aligning the statistical properties of each input series, which helps the Transformer generalize to the whole distribution of data. Both of them play a huge role in our HTPGM. With the incorporation of HTPGM, the model can consider the non-stationary change of real-world time series, leading to improved prediction accuracy for detailed series variation as shown in red circle, which is crucial in real-world time series forecasting tasks.

HTPGM architecture evaluation: To further understand the role of HTPGM in our model, we perform an ablation study examining the impact of the hierarchical time series variational scheme, and the combined optimization objective.

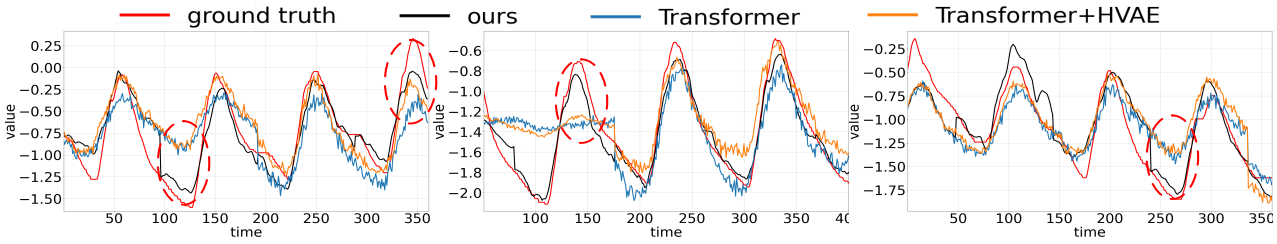


Figure 4: Visualizations on ETTm2 dataset given by different models

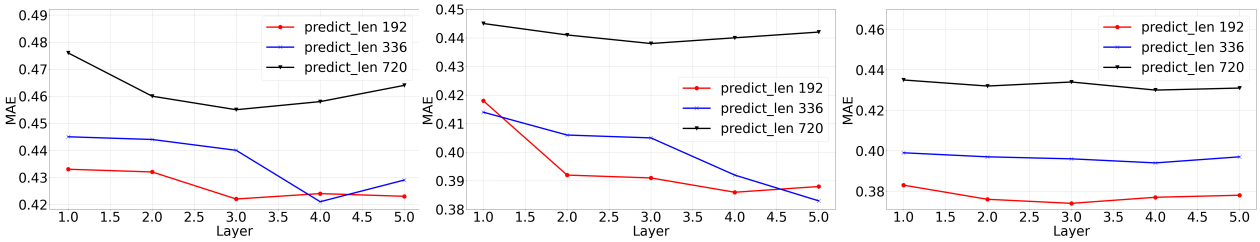


Figure 5: The effectiveness evaluation of hierarchical architecture on ETTh1(left), ETTh2(middle) and ETTm1(right) dataset.

The results on forecasting tasks are presented in Table 2 and Fig. 5. We choose Mean Absolute Error (MAE) to measure the performance of forecasting models in Table 2 and Fig. 5. As shown in the tables, it is obvious that all structural components contribute to the performance of the framework. In particular, the incorporation of a powerful variational generative module for Transformer leads to a significant improvement in performance, which is proved in Table 2, enabling the model to extract robust and non-stationary representations of MTS with complex distribution. The output of transformer h and dynamic prior information modeling into the generative process further improves the generative capacity of the model and leads to better performance on the forecasting tasks. Furthermore, as shown in the second column of Table 2, the ablation study of the combined optimization objective means the latent variable z can learn the effective and expressive representations of time series. Comparison of the five layers results in Fig 5 demonstrates the effectiveness of the hierarchical structure. It is because the different layers include different scale robust and non-stationary information which help the representation to be multi-scale. These results verify the efficacy and necessity of each module in our design.

Balance between Stationary and Non-stationary Information

As discussed in Sec. 3, we combine normalized $\boldsymbol{x}$ and latent representation $\boldsymbol{z}$ of original input on HTV-Trans, which enables our models to recover the non-stationary information efficiently and introduce a parameter α to balance the effect of them. Here, we evaluate the influence of α to MTS forecasting, the results are reported in Fig. 6. As we can see, excessively small and large α will lead to weaker performance, illustrating the effectiveness of the method that we proposed to recover the non-stationary information. It is

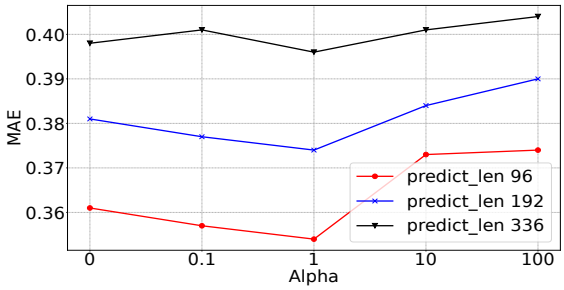


Figure 6: The effectiveness of parameter α on dataset ETTm1 with different predict length.

because when α is smaller the the non-stationary information is lower, and α is bigger means the the non-stationary information is higher. Only the appropriate value of α can bring suitable non-stationary information.

Conclusion

In this paper, we propose a novel approach to solve the over-stationarization problem for multivariate time series forecasting tasks, named HTV-Trans, which consists of a HTPGM module and a transformer that is able to capture non-stationary, non-deterministic, and long-distance temporal dependencies. To achieve efficient optimization, we introduce an autoencoding variational inference scheme with a combined prediction and reconstruction loss. The HTV-Trans model is able to extract robust and intrinsic non-stationary representations of multivariate time series, which allows it to outperform other models in forecasting tasks. Empirical results on MTS forecasting tasks demonstrate the effectiveness of the proposed model.

Acknowledgements

This work was supported in part by the stabilization support of National Radar Signal Processing Laboratory under Grant (JKW202X0X) and National Natural Science Foundation of China (NSFC) (6220010437). The work of Bo Chen acknowledges the support of the National Natural Science Foundation of China under Grant U21B2006; in part by Shaanxi Youth Innovation Team Project; in part by the Fundamental Research Funds for the Central Universities QTZX23037 and QTZX22160; in part by the 111 Project under Grant B18039.

References

Bai, L.; Yao, L.; Kanhere, S. S.; Yang, Z.; Chu, J.; and Wang, X. 2019. Passenger demand forecasting with multi-task convolutional recurrent neural networks. *in PAKDD*, 29–42.

Box, G. E. P. 1976. *Time series analysis, forecasting and control rev. ed*. Time series analysis, forecasting and control rev. ed.

Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. 2020. Spectral temporal graph neural network for multivariate time-series forecasting. *in NeurIPS*, 33: 17766–17778.

Chen, M.; Peng, H.; Fu, J.; and Ling, H. 2021. Autoformer: Searching transformers for visual recognition. *in CVPR*, 12270–12280.

Chen, W.; Chen, B.; Liu, Y.; Wang, C.; Zhou, M.; and Liu, H. 2022. Infinite Switching Dynamic Probabilistic Network with Bayesian Nonparametric Learning. *in TSP*.

Chen, W.; Chen, B.; Liu, Y.; Zhao, Q.; and Zhou, M. 2020. Switching Poisson gamma dynamical systems. *in IJCAI*.

Dai, L.; Chen, W.; Liu, Y.; Argyriou, A.; Liu, C.; Lin, T.; Wang, P.; Xu, Z.; and Chen, B. 2022. Switching Gaussian Mixture Variational RNN for Anomaly Detection of Diverse CDN Websites. *in INFOCOM*.

Dai, L.; Lin, T.; Liu, C.; Jiang, B.; Liu, Y.; Xu, Z.; and Zhang, Z.-L. 2021. SDFVAE: Static and Dynamic Factorized VAE for Anomaly Detection of Multivariate CDN KPIs. *in WWW*, 3076–3086.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *in ICLR*.

Hyndman, R. J.; and Athanasopoulos, G. 2018. *Forecasting: principles and practice*. OTexts.

Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.-H.; and Choo, J. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. *in CVPR*.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2019. Reformer: The efficient transformer. *in ICLR*.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2021. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. *In ICLR*.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. *in NeurIPS*.

Malhotra, P.; Ramakrishnan, A.; Anand, G.; Vig, L.; Agarwal, P.; and Shroff, G. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. *in ICML*.

Ogasawara, E.; Martinez, L. C.; De Oliveira, D.; Zimbrão, G.; Pappa, G. L.; and Mattoso, M. 2010a. Adaptive normalization: A novel data normalization approach for non-stationary time series. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Ogasawara, E.; Martinez, L. C.; Oliveira, D. D.; Silva, G. Z. D.; and Mattoso, M. 2010b. Adaptive Normalization: A novel data normalization approach for non-stationary time series. In *International Joint Conference on Neural Networks, IJCNN 2010, Barcelona, Spain, 18-23 July, 2010*.

Passalis, N.; Tefas, A.; Kanniainen, J.; Gabbouj, M.; and Iosifidis, A. 2019a. Deep adaptive input normalization for time series forecasting. *IEEE transactions on neural networks and learning systems*, 31(9): 3760–3765.

Passalis, N.; Tefas, A.; Kanniainen, J.; Gabbouj, M.; and Iosifidis, A. 2019b. Deep Adaptive Input Normalization for Time Series Forecasting. *Papers*.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *in International Journal of Forecasting*, 36(3): 1181–1191.

Tang, X.; Yao, H.; Sun, Y.; Aggarwal, C.; Mitra, P.; and Wang, S. 2020. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. *in AAAI*, 34(04): 5956–5963.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *in NeurIPS*, 30.

Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; and Sun, L. 2022. Transformers in Time Series: A Survey. *in arXiv preprint arXiv:2202.07125*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; and Li, Z. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. *in AAAI*, 32(1).

Zhang, C.; Song, D.; Chen, Y.; Feng, X.; Lumezanu, C.; Cheng, W.; Ni, J.; Zong, B.; Chen, H.; and Chawla, N. V. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. *in AAAI*, 33(01): 1409–1416.

Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *The Eleventh International Conference on Learning Representations*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. *in AAAI*.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. *In ICML*.