HDT: Hierarchical Discrete Transformer for Multivariate Time Series Forecasting

Shibo Feng[1, 2], **Peilin Zhao**[3*], **Liu Liu**[3], **Pengcheng Wu**[2], **Zhiqi Shen**[1*]

[1]College of Computing and Data Science, Nanyang Technological University (NTU), Singapore
[2]Webank-NTU Joint Research Institute on Fintech, NTU, Singapore
[3]Tencent AI Lab, Shenzhen, China
{shibo001, pengcheng.wu, zqshen}@ntu.edu.sg, {leonliuliu, masonzhao}@tencent.com

Abstract

Generative models have gained significant attention in multivariate time series forecasting (MTS), particularly due to their ability to generate high-fidelity samples. Forecasting the probability distribution of multivariate time series is a challenging yet practical task. Although some recent attempts have been made to handle this task, two major challenges persist: 1) some existing generative methods underperform in high-dimensional multivariate time series forecasting, which is hard to scale to higher dimensions; 2) the inherent high-dimensional multivariate attributes constrain the forecasting lengths of existing generative models. In this paper, we point out that discrete token representations can model high-dimensional MTS with faster inference time, and forecasting the target with long-term trends of itself can extend the forecasting length with high accuracy. Motivated by this, we propose a vector quantized framework called **H**ierarchical **D**iscrete **T**ransformer (HDT) that models time series into discrete token representations with ℓ_2 normalization enhanced vector quantized strategy, in which we transform the MTS forecasting into discrete tokens generation. To address the limitations of generative models in long-term forecasting, we propose a hierarchical discrete Transformer. This model captures the discrete long-term trend of the target at the low level and leverages this trend as a condition to generate the discrete representation of the target at the high level that introduces the features of the target itself to extend the forecasting length in high-dimensional MTS. Extensive experiments on five popular MTS datasets verify the effectiveness of our proposed method. The source code will be released.

Introduction

Multivariate time series forecasting task has been applied to many real-world applications, such as economics (Sezer, Gudelek, and Ozbayoglu 2020; Feng et al. 2022), traffic (Wu et al. 2020; Liu et al. 2016), energy (Zhicheng et al. 2024) and weather (Qiu et al. 2017; Jin et al. 2023). As a generative task, MTS forecasting presents challenges in two key aspects: the inherent high-dimensionality of the data distribution, and the long-term forecasting. To model the complex distributions of high-dimensional data, previous studies have established deep generative models in both autoregressive and non-autoregressive ways. To our knowledge, most of the work in the context of high-dimensional MTS has focused on short-term forecasting (predicted length: 24, 48) (Rasul et al. 2020, 2024; Fan et al. 2024).To improve long-term forecasting, various Transformer architectures (Nie et al. 2022; Liu et al. 2023) have been proposed, but most are focused on low-dimensional scenarios. Effectively modeling high-dimensional distributions with longer forecasting lengths remains a challenge. A key issue is integrating deep generative models with sequence modeling frameworks to handle both high-dimensional data and long-term forecasting tasks.

Existing works (Salinas et al. 2020; Rasul et al. 2021; Li et al. 2022; Feng et al. 2023) have several attempts to utilize various forms of deep generative models, such as Normalizing flows (Dinh, Sohl-Dickstein, and Bengio 2016), Variational Auto-Encoder (VAEs) (Kingma and Welling 2013), Diffusion models (Li et al. 2024; Fan et al. 2024) to model high-dimensional MTS. They apply deep generative models to the high-dimensional distributions over time, learning the patterns of distribution changes along the temporal dimension for precise prediction. Due to complex patterns and long temporal dependencies of MTS, directly modeling high-dimensional MTS distributions in the time domain can lead to issues of distribution drift (Kim et al. 2021) and overlook the correlations between variables, limited to short-term forecasting settings.

Recently, several attention-variant Transformer frameworks (Liu et al. 2023; Rao, Li, and Miao 2022) and LLM-based structures (Zhou et al. 2023; Bian et al. 2024) have been applied to long-term forecasting of MTS, showing excellent performance on MTS datasets. Building on the success of these methods, we identified two key modules: the series decomposition block (Wu et al. 2021; Liu et al. 2022), which uses moving averages to smooth periodic fluctuations and highlight long-term trends, and the discrete Transformer for MTS modeling. Inspired by these approaches, we first learn the discrete representations of the MTS and then incorporate the long-term trends of the forecasting target into our model. This allows us to enhance forecasting length capability with high accuracy.

As a discrete framework, Vector Quantized (Gray 1984) techniques have shown strong competitiveness in high-dimensional image fields (Rao et al. 2021; Zheng et al.

*Corresponding author

2022; Chang et al. 2023), These approaches utilize the pre-quantizing images into discrete latent variables and modeling them autoregressively. For the time series domain, VQ-based methods such as TimeVQVAE (Lee, Malacarne, and Aune 2023), TimeVAE (Desai et al. 2021) and TimeGAN (Yoon, Jarrett, and Van der Schaar 2019) all focus on time series generation task, the lateset VQ-TR (Rasul et al. 2024) introduce the VQ strategy within the transformer architecture as part of the encoder attention blocks, which attends over larger context windows with linear complexity in sequence length for efficient probabilistic forecasting. Inspired by their success of discrete strategy, we aim to explore the application of these techniques in the domain of high-dimensional MTS. Our model differs VQ-TR in two key aspects: i) HDT is two-stage, whereas it is end-to-end. ii) We focus on enhancing the long-term forecasting performance by introducing discrete representation of target itself, while they take efforts to reduce time and space complexity by discretizing the context inputs for efficient forecasting.

To extend the forecasting length within the high-dimensional MTS, we propose an effective generative framework, which is called Hierarchical Discrete Transformer **HDT**. It is a two-stage learning framework, consisting of a pre-quantizing module to obtain the discrete latent tokens of the forecasting targets, called tokenization, and a hierarchical modeling strategy for generating the discrete tokens. In the stage 1, we design two discrete token learning modules: one for obtaining latent tokens of our forecasting targets, and the other for obtaining latent tokens of downsampled targets using the downsampled input. This approach yields two key benefits: i) compressed latent discrete tokens effectively extend the prediction length for high-dimensional MTS, and ii) by incorporating the discrete latent space features of the targets, we reduce time complexity through shorter discrete token generation in stage 2.

In the stage 2, we devise a hierarchical discrete Transformer. At the low-level, we perform cross-attention between the contextual information and the discrete downsampled targets to generation task of downsampling target. At the high-level, we use the discrete downsampled results generated at the low-level as conditions to perform self-conditioned cross-attention with the discrete target, thereby achieving the generation of the discrete target. We summarize our main contributions as follows.

- We propose an effective hierarchical vector quantized method to introduce the long-term trend of targets for future target forecasting with higher accuracy and faster inference time.
- We build a vector quantized MTS framework with ℓ_2 normalization and self-conditioned cross attention for MTS forecasting, which can scale to high-dimensional and extend the prediction length with high accuracy.
- Extensive experiments conducted on real-world datasets demonstrate the superiority of our **HDT**, achieving an average **16.7**% improvement on CRPS_{sum} and **15.4**% on $\text{NRMSE}_{\text{sum}}$, compared to the state-of-the-art methods.

Methods

Our model comprises several key components. In this section, we present an overview of these components, which are divided into two stages. The training and inference details are shown in Algorithm 1, 2 and 3. Figure 1 provides an overview of the model architecture. In the stage 1, we have two types of VQGAN (Esser, Rombach, and Ommer 2021) structures (Encoder, Quantization, Decoder): one is based on the discrete representation learning of the downsampled time series, and the other is based on the discrete representation learning structure corresponding to the prediction targets. Since the VQ strategy is operated on the channel dimension, the inter-variate correlations are captured in stage 1. In stage 2, a context encoder and a base Transformer decoder perform temporal cross-attention to generate discrete downsampled targets. The output from these low-level modules is then fed into a self-conditioned Transformer decoder to autoregressively predict discrete target tokens. This two-stage approach captures inter- and intra-correlations with discrete tokens, enhancing the accuracy of time series forecasting.

Stage 1: Modulating Quantized Vector

Series Downsample Module. According to the Autoformer (Wu et al. 2021), the moving average operation of non-stationary time series can smooth out periodic fluctuations and highlight long-term trends. As the objective of our work is to address the challenge of long-term forecasting in high-dimensional MTS, it is crucial for us to retain long-term patterns with the downsampled time series. For length-τ input series $\mathcal{X}_{pred} \in \mathbb{R}^{\tau \times D}$, the process is:

$$\mathcal{X}_{down} = \text{AvgPool}(\text{Padding}(\mathcal{X}_{pred})), \tag{1}$$

where $\mathcal{X}_{down} \in \mathbb{R}^{\tau \times D}$ denotes the long-term pattern representations. Here, we introduce the $\text{AvgPool}(.)$ for moving average with the $\text{Padding}(.)$ to keep the series length unchanged. $\mathcal{X}_{down}$ is the self-condition of targets, which consists of long-term patterns for the following future targets forecasting.

Discrete Tokenization using VQGAN. In the discrete representation learning of stage 1, the discrete learning modules of targets and downsampled targets show the same structure, which consists of an encoder and a decoder, with a quantization layer that maps a time series input into a sequence of tokens from a learned codebook. The details of these modules are provided in the Appendix C. Specifically, given any time series $\mathcal{X}_{pred} \in \mathbb{R}^{\tau \times D}$ can be represented by a spatial collection of codebook entries $z_{\mathbf{q}_t} \in \mathbb{R}^{s \times n_z}$, where n_z is the dimensionality of quantized vectors in the codebook and s is the length of the discrete token sequence. In this way, each time series can be equivalently represented as a compact sequence with s indices of the code vectors. The quantization operates on the channel dimension, capturing inter-variate correlations. Formally, the observed target $\mathcal{X}_{pred}$ and down-

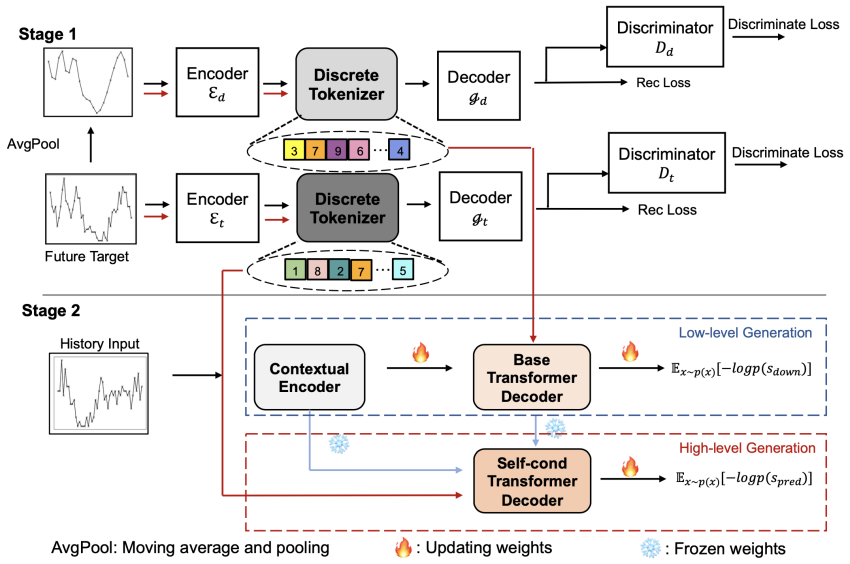

Figure 1: An illustration of our proposed HDT is provided. In stage 1, the model generates discrete downsampled targets and discrete targets, which are passed to Stage 2 for further processing. In stage 2, the contextual encoder and base Transformer decoder are trained with historical inputs and discrete downsampled tokens at the low level. Once trained, these low-level modules are fixed, and their outputs are fed into the high-level framework to generate the final discrete target sequence.

sampled target $\mathcal{X}_{down}$ are reconstructed by:

$$\hat{\mathcal{X}_{pred}} = \mathcal{G}_{\theta_t}\left(z_{\mathbf{q}_t}\right) = \mathcal{G}_{\theta_t}(\mathbf{q}_t(\hat{z}^t)) = \mathcal{G}_{\theta_t}\left(\mathbf{q}_t\left(\mathcal{E}_{\psi_t}(\mathcal{X}_{pred})\right)\right), \tag{2}$$

$$\hat{\mathcal{X}_{down}} = \mathcal{G}_{\theta_d}\left(z_{\mathbf{q}_d}\right) = \mathcal{G}_{\theta_d}(\mathbf{q}_d(\hat{z}^d)) = \mathcal{G}_{\theta_d}\left(\mathbf{q}_d\left(\mathcal{E}_{\psi_d}(\mathcal{X}_{down})\right)\right). \tag{3}$$

In particular, the $\mathcal{E}_{\psi_{[t,d]}}, \mathbf{q}_{[t,d]}, \mathcal{G}_{\theta_{[t,d]}}$ are the encoders, quantization layers and decoders corresponding to $\mathcal{X}_{pred}$ and $\mathcal{X}_{down}$, respectively. To avoid confusion and redundant expressions, we have removed the subscript symbols corresponding to the discrete learning and training process in the stage 1 formulas. The quantization operator $\mathbf{q}$ is conducted to transfer the continuous feature into the discrete space by looking up the closest codebook entry z_k for each timestamp feature $\hat{z}_i$ within $\hat{z}$, and note that $\hat{z}$ represents the execution process corresponding to both $\hat{z}^t$ and $\hat{z}^d$.:

$$z_q = \mathbf{q}(\hat{z}) = \arg\min_{z_k \in \mathcal{Z}} \|\hat{z}_i - z_k\|, \tag{4}$$

where $\mathcal{Z} \in \mathbb{R}^{K \times n_z}$ is the codebook that consists of K entries with n_z dimensions and $\hat{z}_i$ is the continuous feature of the timestamp. Note that $z_{\mathbf{q}_t}$ and $z_{\mathbf{q}_d}$ each correspond to their respective codebooks $\mathcal{Z}^t$ and $\mathcal{Z}^d$. The subscript for $\mathcal{Z}$ is omitted to maintain the brevity of the paper. The above models and the codebook can be learned by optimizing the following objectives:

$$\begin{aligned}\mathcal{L}_{VQ}\left(\mathcal{E}_\psi, \mathcal{G}_\theta, \mathcal{Z}\right) =& \|\mathcal{X} - \hat{\mathcal{X}}\|_2^2 + \|\mathrm{sg}\left[\mathcal{E}_\psi(\mathcal{X})\right] - z_q\|_2^2 \\ &+ \beta \|\mathrm{sg}\left[z_q\right] - \mathcal{E}_\psi(\mathcal{X})\|_2^2.\end{aligned} \tag{5}$$

In detail, sg denotes the stop-gradient operator, β is a hyperparameter for the last term *commitment loss*. The first term is *reconstruction loss* and the second is *codebook loss* to optimize the entries in the codebook.

To learn a perceptually rich codebook in VQGAN, it introduces an adversarial training procedure with a patch-based discriminator D=$\{D_t, D_d\}$ (Isola et al. 2017) that aims to differentiate between real and reconstructed images. In our setting, we introduce a shallow Conv1d network to enhance the reconstruction results:

$$\mathcal{L}_{\mathrm{GAN}}(\{\mathcal{E}_\psi, \mathcal{G}_\theta, \mathcal{Z}\}, D) = [\log D(\mathcal{X}) + \log(1 - D(\hat{\mathcal{X}}))]. \tag{6}$$

The final objective for finding the optimal Model $\mathcal{Q}^* = \mathcal{E}_\psi, \mathcal{G}_\theta, \mathcal{Z}$ is:

$$\begin{aligned}\mathcal{Q}^* = \arg\min_{\mathcal{E}_\psi, \mathcal{G}_\theta, \mathcal{Z}} \max_D \mathbb{E}_{\mathcal{X}\sim p(\mathcal{X})} \big[&\mathcal{L}_{\mathrm{VQ}}(\mathcal{E}_\psi, \mathcal{G}_\theta, \mathcal{Z}) \\ &+ \lambda \mathcal{L}_{\mathrm{GAN}}(\{\mathcal{E}_\psi, \mathcal{G}_\theta, \mathcal{Z}\}, D)\big],\end{aligned}$$

where the λ is an adaptive weight parameter, which is computed by the gradient of $\mathcal{G}_\theta$ and D.

ℓ_2 Regularization. However, in our experiments, we observed that applying $\mathbf{l}_2$ normalization ($\frac{\mathbf{x}}{\|\mathbf{x}\|_2}$) to the entries in the codebook can enhance the reconstruction performance.

$$\mathcal{L}_{norm} = \|\ell_2\left(\mathcal{E}_\psi(\mathcal{X})\right) - \ell_2\left(z_k\right)\|_2^2. \tag{7}$$

Finally, the training loss function is described as:

$$\mathcal{L} = \mathcal{L}_{VQ}\left(\mathcal{E}_\psi, \mathcal{G}_\theta, \mathcal{Z}\right) + \mathcal{L}_{\mathrm{GAN}}(\{\mathcal{E}_\psi, \mathcal{G}_\theta, \mathcal{Z}\}, D) + \mathcal{L}_{norm}. \tag{8}$$

Overall, in the stage 1, $\mathcal{X}_{pred}$ and $\mathcal{X}_{down}$ each obtain their respective codebooks $\mathcal{Z}^t$ and $\mathcal{Z}^d$.

Stage 2: Modelling Prior Distribution with HDT

In this section, we introduce the details of the hierarchical discrete transformer. In stage 2, we establish a framework to

estimate the underlying prior distribution over the discrete space for generating discrete time series tokens. This allows the post-quantization layers and the decoder from stage 1 to reconstruct the continuous targets. First, we present the overall generation process for the discrete tokens, as illustrated in Figure 1. Then, we detail the specific implementation procedures for both the low-level and high-level generation separately.

Low-level Token Generation. This process can be considered a preliminary process of target token generation of high-level. Specifically, we now have the context data $\mathcal{X}_p \in \mathbb{R}^{h\times D}$ and the discrete representation of the downsampled target $s_{down} = \{z_{\mathbf{q}_d}^{s_1}, z_{\mathbf{q}_d}^{s_2}, ..., z_{\mathbf{q}_d}^{s_d}\} \in \mathbb{R}^{s_d\times n_z}$, where h is the look-back window length and D is the number of variates, s_d is the length of discrete downsampled target sequence and n_z is the feature dimension of the discrete representation. We formulate the training process by:

$$\mathcal{H}_p = \mathcal{E}_T(\mathcal{X}_p), \tag{9}$$

$$p(s_{down}|c) = \prod_i p\left(z_{\mathbf{q}_d}^{s_i} \mid z_{\mathbf{q}_d}^{s<i}, c = \mathcal{H}_p\right), \tag{10}$$

$$\mathcal{L}_{\text{base}} = \mathbb{E}_{x\sim p(x)}[-\log p(s_{down})], \tag{11}$$

where $\mathcal{E}_T$ is the contextual encoder that is the Transformer encoder in our experiment. $\mathcal{H}_p \in \mathbb{R}^{h\times n_z}$ is the output of the context encoder and $\mathcal{L}_{\text{base}}$ is the loss function of base Transformer decoder at the low-level framework. $p\left(z_{\mathbf{q}_d}^{s_i} \mid z_{\mathbf{q}_d}^{s<i}, c = \mathcal{H}_p\right)$ is to compute the likelihood of the full representation $p(s_{down}|c) = \prod_i p\left(z_{\mathbf{q}_d}^{s_i} \mid z_{\mathbf{q}_d}^{s<i}, c = \mathcal{H}_p\right)$. We then obtain the trained context embedding $\mathcal{H}_p$ and the downsampled tokens s_{down}. Moreover, the discrete downsampled results directly impact the generation of high-level discrete targets, we explored three different methods for obtaining $\mathcal{H}_p$. These methods are explained in detail in the subsequent experimental section.

High-level Token Generation. After training the context encoder and base Transformer decoder in the low-level framework, we not only capture the content features of the context but also ensure that the discrete downsampled sequences retain long-term patterns. This provides additional conditions related to the target's own features in the high-level framework, thereby enhancing the accuracy of long-term forecasting. We have the discrete target $s_{pred} = \{z_{\mathbf{q}_t}^{s_1}, z_{\mathbf{q}_t}^{s_2}, ..., z_{\mathbf{q}_t}^{s_p}\} \in \mathbb{R}^{s_p\times n_z}$, s_{down} and $\mathcal{H}_p$, where the s_p is the length of discrete target sequence. The process of autoregressively generating s_{pred} can be described as follows:

$$p(s_{pred}|c) = \prod_i p\left(z_{\mathbf{q}_t}^{s_i} \mid z_{\mathbf{q}_t}^{s<i}, c = \{s_{down}, \mathcal{H}_p\}\right), \tag{12}$$

$$\mathcal{L}_{\text{self-cond}} = \mathbb{E}_{x\sim p(x)}[-\log p(s_{pred})], \tag{13}$$

where the s_{down} and $\mathcal{H}^p$ are fixed, the cross-attention of self-conditioned Transformer decoder is operating between the s_{down} and s_{pred}, the temporal cross-attention is introduced to the $\mathcal{H}^p$ and s_{pred}, as shown in Figure 1. After completing the high-level training, we can input the discrete form of the target into the stage 1 decoder $\mathcal{G}_{\theta_t}$ to reconstruct the predicted target. Notably, unlike the popular diffusion models, the VQ discretization strategy effectively avoids the efficiency issues associated with iterative diffusion structures and autoregressive prediction methods.

Algorithm 1: Training of Stage 1

Input: Set of time series targets $\mathcal{X}_{pred}$
Output: Encoder $\mathcal{E}_{\psi_t}$ and $\mathcal{E}_{\psi_d}$, Decoder $\mathcal{G}_{\theta_t}$ and $\mathcal{G}_{\theta_d}$, Discriminator D_t and D_d, quantization codebook $\mathbf{q}_t$ and $\mathbf{q}_d$.

for $k \leftarrow 1$ to K **do**
  Get the $\text{X}_{\text{pred}} \sim \mathcal{X}_{pred}$;
  Obtain the X_{down} by Eqn. 1;
  Feed X_{pred} and X_{down} to encoder $\{\mathcal{E}_{\psi_t}, \mathcal{E}_{\psi_d}\}$, and quantization $\{\mathbf{q}_t, \mathbf{q}_d\}$, by Eqn.(2, 3, 4), respectively;
  Compute the ℓ_2 Regularization and loss by Eqn.(5, 7);
  if $\text{k} \geq \hat{k}$ is $0.75K$ **then**
    Introduce the Discriminator D_t, D_d respectively and compute the loss by Eqn. 8;
  end if
end for
Return trained $\mathcal{E}_{\psi_t}$, $\mathcal{E}_{\psi_d}$, $\mathcal{G}_{\theta_t}$, $\mathcal{G}_{\theta_d}$, $\mathbf{q}_t$, $\mathbf{q}_d$, D_t and D_d.

Algorithm 2: Training of Stage II

Input: Set of history time series $\mathcal{X}_p$, targets X_{pred} and trainable BOS token [BOS]. The optimized encoders $\mathcal{E}_{\psi_d}$ and $\mathcal{G}_{\theta_t}$, trained quantization codebooks $\mathbf{q}_t$ and $\mathbf{q}_d$.
Output: The base Transformer decoder $\mathcal{B}$, contextual encoder $\mathcal{E}_T$, and self-cond Transformer decoder$\mathcal{S}$.

for $k \leftarrow 1$ to K **do**
  Obtain the X_{down} from $X_{pred}$by Eqn. 1;
  Get the token sequences s_{down} and s_{pred} from trained $\mathbf{q}_t$ and $\mathbf{q}_d$ of stage 1 by Eqn. 4 with X_{down} and X_{pred}, respectively;
  Minimize the negative log-likelihood with training $\mathcal{E}_T$ and $\mathcal{B}$ by Eqn.(9, 10, 11) with concatenating the [BOS] token at the beginning of token sequence s_{down}.
end for
for $k \leftarrow 1$ to K **do**
  Introduce the output s_{down} from the combination of trained $\mathcal{E}_T$ and $\mathcal{B}$;
  Minimize the negative log-likelihood with frozen $\mathcal{E}_T$, $\mathcal{B}$ and trainable $\mathcal{S}$ by Eqn. 13 with concatenating the [BOS] token at the beginning of token sequence s_{pred}.
end for
Return trained contextual encoder $\mathcal{E}_T$, base Transformer decoder $\mathcal{B}$, self-cond Transformer decoder $\mathcal{S}$ and [BOS] token.

Experiments

We conducted experiments to evaluate the performance and efficiency of HDT, covering short-term and long-term forecasting as well as robustness to missing values. The evaluation includes 5 real-world benchmarks and 12 baselines.

Algorithm 3: Inference

Input: Set of history time series $\mathcal{X}_p$, trained BOS token [BOS], trained contextual encoder $\mathcal{E}_T$, base Transformer decoder $\mathcal{B}$ and self-cond Transformer $\mathcal{S}$ and Decoder $\mathcal{G}_{\theta_t}$.
Output: Reconstructed future targets $\mathcal{X}_{pred}$.

1: **for** $i \leftarrow 1$ to I in test samples **do**
2: Sample the downsampled tokens s_{down} with trained [BOS] token and $\mathcal{X}_p$ from the combination of $\mathcal{E}_T$ and $\mathcal{B}$ by Eqn.(9, 10, 11);
3: Sample the target tokens s with [BOS] token from trained $\mathcal{S}$, $\mathcal{E}_T$ and $\mathcal{B}$ by Eqn.(12, 13);
4: Return the target X_{pred} by $\mathcal{G}_{\theta_t}$ of Eqn.2.
5: **end for**
6: Return the prediction target $\mathcal{X}_{pred}$.

Detailed model and experiment configurations are summarized in Appendix C.

Datasets. We extensively evaluate the proposed HDT on five real-world benchmarks, covering the mainstream high-dimensional MTS probabilistic forecasting applications, Solar (Lai et al. 2018), Electricity (Lai et al. 2018), Traffic (Salinas et al. 2019), Taxi (Salinas et al. 2019) and Wikipedia (Gasthaus et al. 2019). These data are recorded at intervals of 30 minutes, 1 hour, and 1 day frequencies, more details refer to Appendix B.

Baselines. We include several competitive multivariate time series baselines to verify the effectiveness of HDT. Previous work DeepAR (Salinas et al. 2020), GP-Copula (Salinas et al. 2019) and Transformer-MANF (Rasul et al. 2020). Then, we compare HDT against the diffusion-based methods, TimeGrad (Rasul et al. 2021), MG_TSD (Fan et al. 2024), D^3VAE (Li et al. 2022), CSDI (Tashiro et al. 2021), SSSD (Alcaraz and Strodthoff 2022), TSDiff (Kollovieh et al. 2023) with additional Transformer layers followed by S4 layer and TimeDiff (Shen and Kwok 2023). Among the MTS forecasting with VQ-Transformer, we introduce and VQ-TR (Rasul et al. 2023) for comparisons. The details of baselines are shown in Appendix F.

Evaluation Metrics. For probabilistic estimates, we report the continuously ranked probability score across summed time series (CRPS_{sum}) (Matheson and Winkler 1976), a widely used metric for probabilistic time series forecasting, as well as a deterministic estimation metric NRMSE_{sum} (Normalized Root Mean Squared Error). For detailed descriptions, refer to Appendix B.

Implementation Details. Our method relies on the ADAM optimizer with initial learning rates of 0.0005 and 0.001, and a batch size of 64 across all datasets. The history length is fixed at 96, with prediction lengths of $\{48, 96, 144\}$. We sample 100 times to report metrics on the test set. All experiments are conducted on a single Nvidia A-100 GPU, and results are based on 3 runs.

Main results

Probabilistic Forecasting Performance. As shown in Table 1, HDT achieves consistent state-of-the-art performance in most of benchmarks, covering three prediction settings, large span of dimensions and more showcases are shown in Supplementary due to the page limitation. Especially, HDT achieves a large performance gain over recent popular discrete method VQ-TR, such as average **25.9**% CRPM_{sum} improvement on Traffic, **23.1**% CRPM_{sum} improvement on Taxi. Also, we observe that HDT outperforms some diffusion-based methods TimeDiff, TSDiff and marginal improvement against recent strong baseline MG_TSD that is more obvious in the case of high-dimensional and non-stationary datasets, such as average **13.3**% improvement on Traffic and **10.2**% on Taxi. This implies that the trends of target may introduce more future information gains into our forecasting model.

Deterministic Forecasting Performance. In our experiments, we observe that some models exhibit higher values for CRPS_{sum}, yet lack true predictive accuracy. Therefore, we report the NRMSE_{sum} for deterministic estimation, which is shown in Table 1. We found that HDT achieves the best results cross all datasets, especially in the Traffic and Taxi datasets, we achieve average improvements of **15.3**% and **15.6**% NRMSE_{sum} comparing to the strong baseline MG_TSD. It is worth noting that D^3VAE and TimeDiff showe significant deviations in point evaluations, but TSDiff with self-guidance demonstrate competitive performance, implying the effectiveness of self-guided strategy. The time and space efficiencies of HDT are shown in Appendix D, due to limited space.

Ablation studies

Effect of Discrete Representation z_q in Eqn. (4). To verify the effectiveness of discrete representations in MTS, we conducted an experiment by bypassing the discretization of the intermediate variable $\hat{z}$ in stage 1, directly inputting it into stage 2 for autoregressive generation via cross-attention with the context encoder. We tested this on three datasets (Electricity, Traffic, Taxi) with two prediction lengths (48 and 96), covering dimensions from 370 to 1214. As shown in Table 2, the continuous structure (C-Transformer) performed poorly in both probabilistic and deterministic scenarios. We believe that without discretization, $\hat{z}$ acts as an infinitely large codebook, making it difficult for the stage 2 Transformer to fit properly. This highlights the effectiveness of our discrete Transformer structure.

Effect of Historical Condition $\mathcal{H}^p$ in Eqn. (13). To verify the applicability of discrete representations in multivariate time series, we set four different forms of historical sequences during the second stage of training: (i) HDT-h_c: the continuous features from the stage1, not transformed into discrete form; (ii) HDT-h_d: transformed into the corresponding discrete form in the stage 1; (iii) HDT-h_{d^*}: the discrete features, without entering the Encoder of stage 2 (iv) HDT-h_{dc}: concatenation of discrete and continuous representations from the stage 1. We test on two high-dimensional and distinct types of multivariate time series and the results are shown in Table 3, the relatively stable and periodic Traffic, and the Taxi series, which is of higher frequency of fluctuations and more outliers. We observe that in the Traffic and Taxi of all prediction settings, HDT-h_c performs obviously lower than HDT-h_d and HDT-h_{d^*}, while HDT-h_{dc} is com-

Models		**HDT (Ours)**		VQ-TR (2024)		MG_TSD (2024)		TSDiff (2023)		TimeDiff (2023)		SSSD (2023)		D^3VAE (2022)		CSDI (2021)		TimeGrad (2021)		Trans-MAF (2020)		DeepAR (2020)		GP-Copula (2019)	
Metric		$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE	$\hat{\text{CRPS}}$	NRMSE
Solar	48	0.329	0.653	0.334	0.657	0.328	**0.645**	**0.324**	0.651	0.376	0.814	0.340	0.654	0.382	0.692	0.336	0.651	0.357	0.667	0.341	0.672	0.362	0.691	0.426	0.891
	96	**0.330**	**0.694**	0.357	0.734	0.339	0.707	0.336	0.715	0.415	0.935	0.365	0.704	0.413	0.757	0.359	0.712	0.384	0.731	0.376	0.743	0.402	0.775	0.475	0.921
	144	**0.357**	**0.776**	0.377	0.885	0.373	0.825	0.379	0.847	0.438	1.312	0.392	0.830	0.448	0.914	0.387	0.865	0.429	0.916	0.394	0.824	0.448	0.936	0.559	1.207
	Avg	**0.338**	**0.707**	0.356	0.758	0.347	0.726	0.346	0.738	0.410	1.020	0.366	0.729	0.414	0.788	0.361	0.743	0.390	0.771	0.370	0.746	0.404	0.801	0.487	1.006
Electricity	48	0.025	0.030	0.034	0.033	**0.023**	0.030	0.024	**0.029**	0.036	0.092	0.037	0.032	0.046	0.096	0.032	0.034	0.043	0.031	0.039	0.034	0.043	0.035	0.047	0.055
	96	**0.028**	**0.032**	0.045	0.040	0.034	0.035	0.039	0.036	0.049	0.109	0.045	0.041	0.062	0.114	0.049	0.039	0.067	0.035	0.060	0.038	0.058	0.044	0.069	0.058
	144	**0.036**	**0.057**	0.049	0.070	0.042	0.064	0.047	0.072	0.063	0.147	0.056	0.084	0.086	0.142	0.067	0.088	0.082	0.085	0.101	0.093	0.104	0.097	0.125	0.109
	Avg	**0.028**	**0.038**	0.043	0.048	0.033	0.043	0.036	0.046	0.049	0.116	0.046	0.052	0.065	0.117	0.049	0.054	0.064	0.050	0.066	0.055	0.068	0.059	0.080	0.074
Traffic	48	**0.034**	**0.060**	0.039	0.074	0.036	0.067	0.057	0.070	0.064	0.175	0.053	0.074	0.082	0.312	-	-	0.067	0.072	0.070	0.074	0.069	0.081	0.082	0.136
	96	**0.037**	**0.063**	0.052	0.082	0.042	0.072	0.068	0.076	0.081	0.246	0.069	0.080	0.091	0.465	-	-	0.095	0.087	0.086	0.081	0.099	0.128	0.093	0.148
	144	**0.047**	**0.076**	0.068	0.093	0.056	0.096	0.095	0.114	0.109	0.304	0.084	0.106	0.129	0.472	-	-	0.124	0.105	0.107	0.096	0.113	0.142	0.125	0.185
	Avg	**0.039**	**0.066**	0.053	0.083	0.045	0.078	0.073	0.087	0.085	0.241	0.069	0.087	0.101	0.416	-	-	0.095	0.088	0.088	0.084	0.093	0.117	0.100	0.156
Taxi	48	**0.166**	**0.264**	0.274	0.363	0.217	0.327	0.243	0.330	0.272	0.391	0.234	0.338	0.246	0.617	-	-	0.264	0.348	0.236	0.345	0.259	0.368	0.276	0.388
	96	**0.356**	**0.513**	0.473	0.577	0.379	0.528	0.469	0.534	0.491	0.590	0.371	0.542	0.481	0.849	-	-	0.488	0.571	0.464	0.563	0.476	0.607	0.617	0.625
	144	**0.465**	**0.538**	0.536	0.724	0.485	0.703	0.517	0.706	0.532	0.915	0.483	0.712	0.527	1.124	-	-	0.515	0.717	0.522	0.726	0.559	0.774	0.664	0.815
	Avg	**0.329**	**0.438**	0.428	0.555	0.360	0.519	0.410	0.523	0.432	0.632	0.363	0.531	0.418	0.863	-	-	0.422	0.545	0.407	0.545	0.431	0.583	0.519	0.609
Wikipedia	48	0.073	0.095	**0.063**	**0.086**	0.066	0.093	0.074	0.090	0.091	0.142	0.077	0.103	0.112	1.625	-	-	0.081	0.102	0.084	0.111	0.083	0.109	0.092	0.107
	96	**0.074**	**0.126**	0.086	0.153	0.080	0.137	0.086	0.143	0.116	0.191	0.093	0.146	0.187	2.234	-	-	0.119	0.194	0.107	0.148	0.105	0.163	0.131	0.160
	Avg	**0.073**	**0.110**	0.074	0.120	**0.073**	0.115	0.080	0.116	0.104	0.167	0.085	0.125	0.150	1.929	-	-	0.100	0.148	0.095	0.130	0.094	0.136	0.111	0.135
1st Count		**16**	**16**	1	1	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 1: Model performance comparisons on the test set $\hat{\text{CRPS}}$:CRPS_{sum}, $\hat{\text{NRMSE}}$:NRMSE_{sum} (lower is better) show baselines and our HDT model. – marks out-of-memory failures. Trans-MAF stands for Transformer-MAF. The underlined ones as the second best.

Datasets	Electricity				Traffic				Taxi			
Lengths	48		96		48		96		48		96	
Metrics	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}
C-Transformer	0.327(.004)	0.532(.018)	0.212(.007)	0.228(.014)	0.467(.011)	0.845(.009)	1.004(.005)	1.150(.012)	0.861(.006)	1.121(.013)	0.977(.008)	1.118(.011)
HDT	0.025(.002)	0.030(.002)	0.028(.001)	0.032(.003)	0.034(.001)	0.060(.004)	0.037(.003)	0.063(.005)	0.166(.005)	0.264(.003)	0.356(.002)	0.513(.007)

Table 2: Performance of HDT with Continuous Transformer structure C-Trasformer, which does not include the quantization layer in the stage 1 and replace the discrete token sequences of HDT with continuous representation from stage 1.

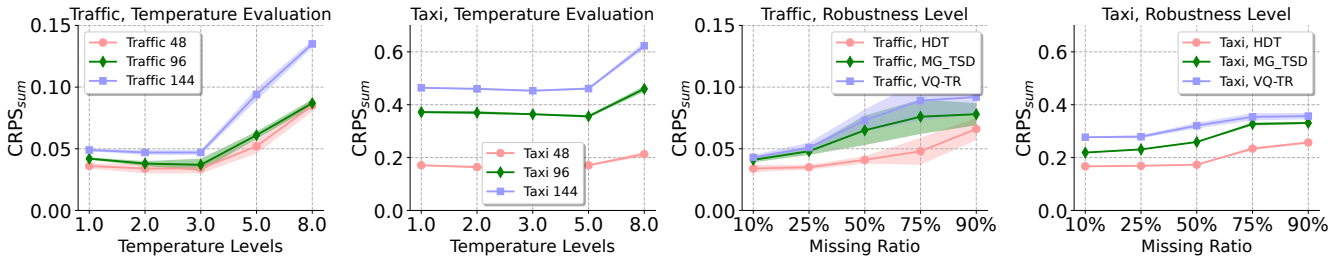

Figure 2: Performance of HDT with different temperature levels of different prediction lengths in Traffic and Taxi datasets. The comparison results against MG_TSD and VQ-TR with HDT on different levels of missing rate.

Datasets	Traffic						Taxi					
Lengths	48		96		144		48		96		144	
Metrics	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}
HDT-h_c	0.042(.003)	0.069(.005)	0.046(.002)	0.077(.006)	0.056(.002)	0.084(.003)	0.206(.004)	0.316(.005)	0.373(.006)	0.548(.010)	0.481(.004)	0.573(.006)
HDT-h_{d^*}	0.038(.002)	0.065(.008)	0.039(.004)	0.067(.004)	0.050(.004)	0.078(.003)	0.189(.004)	0.278(.004)	0.371(.004)	0.537(.005)	0.480(.005)	0.568(.011)
HDT-h_d	**0.034(.001)**	**0.060(.004)**	0.037(.003)	**0.063(.005)**	0.048(.005)	0.079(.006)	**0.166(.003)**	**0.264(.005)**	**0.356(.002)**	**0.513(.004)**	0.467(.006)	0.540(.007)
HDT-h_{dc}	0.036(.003)	0.062(.006)	**0.037(.001)**	0.064(.002)	**0.047(.004)**	**0.076(.008)**	0.171(.005)	0.266(.003)	0.356(.004)	0.517(.009)	**0.465(.003)**	**0.537(.008)**

Table 3: Performance of HDT with Different Types of contextual conditions.**Bold** numbers represent the best outcomes and the underlined ones as the second best.

petitive. This suggests that within a probabilistic framework, discrete representations, serving as an approximate expression, can be seen as a "Clustering" result that is more resilient to stochastic changes. By incorporating target trends,

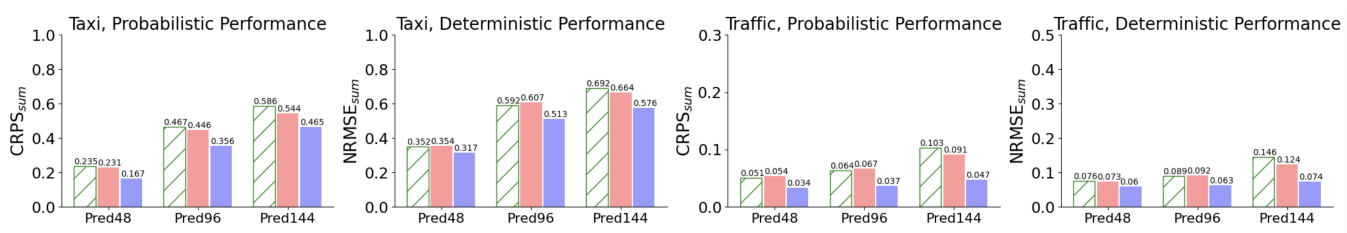

Figure 3: Probabilistic and deterministic performance of HDT and HDT-variants on different prediction length and datasets. HDT-var.T (**Green** bar) is the same structure with HDT (**Purple** bar) without the self-conditions in stage 2. HDT-var.L (**Red** bar) replaces the Transformer with LSTM in stage 2 and without self-conditions.

Datasets	Traffic						Taxi					
Lengths	48		96		144		48		96		144	
Metrics	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}	CRPS_{sum}	NRMSE_{sum}
2	0.036(.003)	0.070(.005)	0.044(.002)	0.079(.007)	0.061(.004)	0.094(.008)	0.226(.005)	0.338(.006)	0.373(.007)	0.530(.012)	0.542(.005)	0.714(.012)
3	0.036(.005)	0.073(.003)	**0.037(.003)**	**0.063(.005)**	0.049(.003)	0.080(.006)	**0.166(.003)**	**0.264(.005)**	0.378(.005)	0.569(.009)	0.530(.008)	0.635(.014)
4	**0.034(.001)**	**0.060(.004)**	0.039(.004)	0.067(.006)	**0.047(.004)**	**0.076(.008)**	0.172(.002)	0.277(.007)	**0.356(.002)**	**0.513(.004)**	**0.465(.003)**	**0.537(.008)**
5	0.037(.002)	0.075(.003)	0.363(.003)	0.524(.005)	0.052(.002)	0.082(.005)	0.173(.003)	0.268(.004)	0.363(.004)	0.526(.007)	0.476(.007)	0.578(.011)

Table 4: Performance of HDT with Transformer layers under different prediction lengths on Traffic (Stationary) and Taxi (Non-stationary). We report mean&stdev.results of 3 runs.

HDT can achieve a higher level of deterministic forecasting performance.

Effect of Discrete Self-Condition s_{down} in Eqn. (10). From Figure 3, we have: i) For the short-term prediction length (e.g.48) of two datasets, both HDT-var.T and HDT-var.L show marginal differences between HDT, implying the effectiveness of discrete representations. In contrast, these variants show obvious differences between HDT of 96 and 144 settings, which further verifies the merits of our self-conditioned strategy.ii) Discrete features demonstrate stable performance in relatively steady dataset(e.g.Traffic), without significant declines as the forecasting horizon extends. However, in non-stationary dataset(e.g.Taxi), it still exhibits notable performance fluctuations of discrete representations, which implies the effectiveness of our self-condition strategy.

Effect of Missing Ratios in Eqn. (13). To evaluate HDT's robustness, we implemented a timestamp masking strategy, allowing the network to infer representations under incomplete contexts. We randomly masked observations (historical sequences) in the test sets of the Traffic (pred 96) and Taxi (pred 48) datasets at designated missing rates. Figure 2 illustrates that excluding the target condition from the forecasting model leads to a rapid decline in probabilistic performance as the missing rate increases in two diffusion models. From the Taxi dataset, with the missing rate of historical conditions nearing 100%, HDT's performance remains largely unaffected, in contrast to the obvious performance degradation observed in the other two history-conditioned diffusion models.

Effect of Temperature Levels in Inference. During our experiments, we observed that sampling temperature is a crucial hyperparameter in a probabilistic setting. As shown in Figure 2, tests on the Traffic and Taxi datasets revealed significant differences in results with varying temperatures. As for the Traffic dataset, a slightly higher temperature improved probabilistic forecasting performance, while a substantial increase led to model bias. For the Taxi dataset, we found that a moderate temperature is optimal, with no significant change in short-term accuracy at higher temperatures compared to long-term settings. This suggests that HDT can achieve better results by adjusting temperature variations to suit different datasets and forecasting lengths.

Effect of Number of Layers in Eqn. (13). To investigate the effect of the self-cond Transformer layers in Eqn. (13), we report the CRPS_{sum} and NRMSE_{sum} results of our SDT with different number of layers (e.g.2, 3, 4, 5) in Table 4. We observe that in short-term forecasting, a smaller number of layers (e.g., 2, 3) shows competitive results in both datasets. As the forecast length increases, Traffic exhibits superior performance with a moderately increased number of layers, while high-stochastic Taxi excels in deeper Transformer structures. These experimental results were all conducted under the condition that the base Transformer decoder layers in Eqn. 11 are fixed at 3.

Conclusion

In this paper, we propose a hierarchical self-conditioned discrete method **HDT** to enhance high-dimensional multivariate time series (MTS) forecasting. Our novel two-stage vector quantized generative framework maps targets into discrete token representations, capturing target trends for long-term forecasting. To the best of our knowledge, this is the first discrete Transformer architecture applied to high-dimensional, long-term forecasting tasks. Extensive experiments on benchmark datasets demonstrate the effectiveness of our approach. Future research will explore integrating multimodal data into MTS forecasting.

Acknowledgments

This research is supported by the Joint NTU-WeBank Research Centre on Fintech, Nanyang Technological University, Singapore.

References

Alcaraz, J. M. L.; and Strodthoff, N. 2022. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint arXiv:2208.09399*.

Bian, Y.; Ju, X.; Li, J.; Xu, Z.; Cheng, D.; and Xu, Q. 2024. Multi-patch prediction: Adapting llms for time series representation learning. *arXiv preprint arXiv:2402.04852*.

Chang, H.; Zhang, H.; Barber, J.; Maschinot, A.; Lezama, J.; Jiang, L.; Yang, M.-H.; Murphy, K.; Freeman, W. T.; Rubinstein, M.; et al. 2023. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*.

Desai, A.; Freeman, C.; Wang, Z.; and Beaver, I. 2021. Timevae: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12873–12883.

Fan, X.; Wu, Y.; Xu, C.; Huang, Y.; Liu, W.; and Bian, J. 2024. MG-TSD: Multi-Granularity Time Series Diffusion Models with Guided Learning Process. *arXiv preprint arXiv:2403.05751*.

Feng, S.; Miao, C.; Xu, K.; Wu, J.; Wu, P.; Zhang, Y.; and Zhao, P. 2023. Multi-scale attention flow for probabilistic time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

Feng, S.; Xu, C.; Zuo, Y.; Chen, G.; Lin, F.; and XiaHou, J. 2022. Relation-aware dynamic attributed graph attention network for stocks recommendation. *Pattern Recognition*, 121: 108119.

Gasthaus, J.; Benidis, K.; Wang, Y.; Rangapuram, S. S.; Salinas, D.; Flunkert, V.; and Januschowski, T. 2019. Probabilistic forecasting with spline quantile function RNNs. In *The 22nd international conference on artificial intelligence and statistics*, 1901–1910. PMLR.

Gray, R. 1984. Vector quantization. *IEEE Assp Magazine*, 1(2): 4–29.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Jin, M.; Koh, H. Y.; Wen, Q.; Zambon, D.; Alippi, C.; Webb, G. I.; King, I.; and Pan, S. 2023. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. *arXiv preprint arXiv:2307.03759*.

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 *International Conference on Learning Representations*.

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

Kollovieh, M.; Ansari, A. F.; Bohlke-Schneider, M.; Zschiegner, J.; Wang, H.; and Wang, Y. 2023. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. *arXiv preprint arXiv:2307.11494*.

Kollovieh, M.; Ansari, A. F.; Bohlke-Schneider, M.; Zschiegner, J.; Wang, H.; and Wang, Y. B. 2024. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. *Advances in Neural Information Processing Systems*, 36.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 95–104.

Lee, D.; Malacarne, S.; and Aune, E. 2023. Vector Quantized Time Series Generation with a Bidirectional Prior Model. *arXiv preprint arXiv:2303.04743*.

Li, Y.; Chen, W.; Hu, X.; Chen, B.; Zhou, M.; et al. 2024. Transformer-Modulated Diffusion Models for Probabilistic Multivariate Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Li, Y.; Lu, X.; Wang, Y.; and Dou, D. 2022. Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35: 23009–23022.

Liu, C.; Hoi, S. C.; Zhao, P.; and Sun, J. 2016. Online arima algorithms for time series prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35: 9881–9893.

Matheson, J. E.; and Winkler, R. L. 1976. Scoring rules for continuous probability distributions. *Management science*, 22(10): 1087–1096.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Qiu, M.; Zhao, P.; Zhang, K.; Huang, J.; Shi, X.; Wang, X.; and Chu, W. 2017. A short-term rainfall prediction model using multi-task convolutional neural networks. In *2017 IEEE international conference on data mining (ICDM)*, 395–404. IEEE.

Rao, H.; Li, Y.; and Miao, C. 2022. Revisiting k-reciprocal distance re-ranking for skeleton-based person re-identification. *IEEE Signal Processing Letters*, 29: 2103–2107.

Rao, H.; Xu, S.; Hu, X.; Cheng, J.; and Hu, B. 2021. Multi-Level Graph Encoding with Structural-Collaborative Relation Learning for Skeleton-Based Person Re-Identification. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 973–980. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Rasul, K.; Bennett, A.; Vicente, P.; Gupta, U.; Ghonia, H.; Schneider, A.; and Nevmyvaka, Y. 2023. VQ-TR: Vector Quantized Attention for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Rasul, K.; Bennett, A.; Vicente, P.; Gupta, U.; Ghonia, H.; Schneider, A.; and Nevmyvaka, Y. 2024. VQ-TR: Vector Quantized Attention for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

Rasul, K.; Seward, C.; Schuster, I.; and Vollgraf, R. 2021. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, 8857–8868. PMLR.

Rasul, K.; Sheikh, A.-S.; Schuster, I.; Bergmann, U.; and Vollgraf, R. 2020. Multivariate probabilistic time series forecasting via conditioned normalizing flows. *arXiv preprint arXiv:2002.06103*.

Salinas, D.; Bohlke-Schneider, M.; Callot, L.; Medico, R.; and Gasthaus, J. 2019. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems*, 32.

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

Sezer, O. B.; Gudelek, M. U.; and Ozbayoglu, A. M. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90: 106181.

Shen, L.; and Kwok, J. 2023. Non-autoregressive Conditional Diffusion Models for Time Series Prediction. *arXiv preprint arXiv:2306.05043*.

Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34: 24804–24816.

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.

Wu, S.; Xiao, X.; Ding, Q.; Zhao, P.; Wei, Y.; and Huang, J. 2020. Adversarial sparse transformer for time series forecasting. *Advances in neural information processing systems*, 33: 17105–17115.

Yoon, J.; Jarrett, D.; and Van der Schaar, M. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32.

Zheng, C.; Vuong, T.-L.; Cai, J.; and Phung, D. 2022. Movq: Modulating quantized vectors for high-fidelity image generation. *Advances in Neural Information Processing Systems*, 35: 23412–23425.

Zhicheng, C.; SHIBO, F.; Zhang, Z.; Xiao, X.; Gao, X.; and Zhao, P. 2024. SDformer: Similarity-driven Discrete Transformer For Time Series Generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36: 43322–43355.