Probabilistic Time Series Forecasting via Diffusion Models under Multi-Scale Guidance

Bowen Pang[a,b], Liyi Huang[b,c], Xiangnan Feng[d], Wei Wei[a,b,c,e,*]

[a] *School of Mathematical Sciences, Beihang University, Beijing, 100191, China*
[b] *Key Laboratory of Mathematics Informatics Behavioral Semantics, Ministry of Education, Beijing, 100191, China*
[c] *School of Artificial Intelligence, Beihang University, Beijing, 100191, China*
[d] *Complexity Science Hub, Vienna, 1080, Austria*
[e] *Zhongguancun Laboratory, Beijing, 100094, China*

Abstract

Diffusion models, due to their outstanding generation capability, have been recently introduced into probabilistic time series forecasting field to estimate the future values of time series in a conditional generation manner, with past observations serving as the condition to provide guidance to the denoising process. In existing approaches, past observations are typically made to apply guidance on a constant time scale throughout the entire process. However, we discover that there exists a *scale transition* phenomenon, suggesting that information on different time scales needs to be highlighted at different denoising stages. Furthermore, an effective guidance mechanism also requires more efficient representation of past observations to contribute to future value generation, which is challenged by the multi-scale entanglement inherent in time series data. In this paper, we propose a novel model, named **M**ulti-sc**A**le **G**uidance **NET**work (**MAGNET**), to solve the problems above, which is based on diffusion model framework and equipped with a uniquely designed *multi-scale guidance*. With this guidance, MAGNET guides the denoising process with consideration of the scale hierarchy required by different stages, better represents historical information through multi-scale feature learning, and makes the denoising more stable, all of which benefit the final forecast-

[*] Corresponding author.

Email addresses: pangbw@buaa.edu.cn (Bowen Pang), liyihuang@buaa.edu.cn (Liyi Huang), fengxiangnan@gmail.com (Xiangnan Feng), weiw@buaa.edu.cn (Wei Wei)

Preprint submitted to Engineering Applications of Artificial Intelligence October 28, 2024

ing. We test MAGNET on both synthetic and real-world datasets. The main experimental results demonstrate the outperformance of MAGNET with multi-scale guidance compared to existing approaches under different evaluation criteria.

Keywords: Time Series Forecasting, Diffusion Models

1. Introduction

Time series is ubiquitous in the real world, which refers to any data organized or collected in chronological order. How to forecast time series accurately is a classic problem, and is also crucial to many domains, such as economics [1], transportation [2, 3], environmental sciences [4], etc. While most existing works focus on deterministic forecasting that generates a single trajectory for future time points, probabilistic forecasting can provide a more comprehensive view of the future situation of a time series by considering the uncertainty of a time series and estimating the distributions for its future values. This distribution-targeted modeling, arguably to be more reasonable and powerful, is especially emphasized when it comes to issues like financial risk management [5], healthcare [6], energy forecasting [7], etc.

As probabilistic forecasting can be taken as a conditional generation task in the view of deep learning, generative models like VAEs (variational autoencoders) [8], GANs (generative adversarial networks) [9], and diffusion models [10, 11], along with their variants, are continuously introduced into this topic, indeed achieving some encouraging results. Among them, diffusion models, characterized by a combination of diffusion and denoising processes, have received considerable attention in recent years due to their outstanding performances in computer vision. When applying them to time series forecasting, the research paradigm is to adopt the past observed sequence as a condition to provide guidance to the denoising process for future sequence generation.

Figure 1 illustrates the process of a time series sample being corrupted to white noise during the diffusion process of diffusion models. It can be observed that features of the time series on large time scales, such as long trends over certain periods, are the first to degrade, followed by those on small scales. This indicates a phenomenon that we refer to as *scale transition*, where time series' features on different time scales are selectively emphasized at different stages of the inner processes of diffusion models. Thus, during the denoising process, if the condition can be made to adaptively provide information on appropriate time scales based on the needs of different stages, the entire process is expected to be more structured and efficient. To the best of our knowledge, no existing diffusion-model-based approaches have ever considered this phenomenon, where the past sequence typically functions by applying guidance on a constant time scale regardless of the varying stages, similar to the classifier [12] and classifier-free [13] guidance used in image generation.

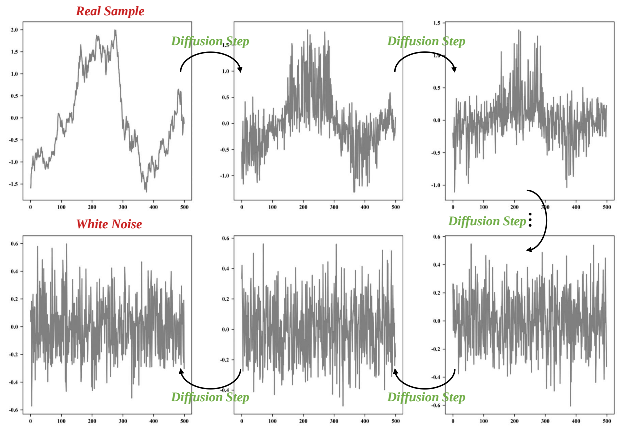


Figure 1: An illustration showing how a time series sample evolves during the diffusion process of diffusion models. The sample is sliced from Exchange[14] dataset.

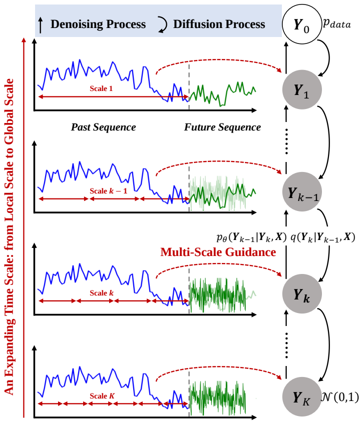


Figure 2: The schematic diagram of **multi-scale guidance**.

In addition, it should also be noted that multi-scale complexity is inherent in time series data, where features on small time scales often correspond to the intrinsic patterns, while features on large time scales are largely determined by some exogenous factors. A typical example is the price of an asset, where short-term trading behaviors of investors drive the short-term movements of the price, while the price patterns over months or years are often influenced by macroeconomic factors like interest environment, fiscal policy, etc. This multi-scale complexity poses great challenges for time series representation that is, however, rather important for an effective guidance mechanism since forecasting heavily relies on the features or patterns extracted from the past observations, i.e., the conditions. In this regard, it is required for a desirable guidance mechanism to disentangle and learn the multi-scale features from the past sequence, and to provide them to denoising for better results.

Under the motivation discussed above, we propose the idea of **multi-scale guidance**, whose schematic diagram is illustrated in Figure 2. With this guidance, the past sequence is made to self-adjust to provide information from local to global time scales to guide the denoising process for future sequence generation, in response to the scale transition phenomenon. Based on this, we further propose a novel model, named **M**ulti-sc**A**le **G**uidance **NET**work (**MAGNET**), to make probabilistic time series forecasting, which possesses a diffusion-model-type architecture and is equipped with the uniquely designed multi-scale guidance. Given the partitioned past and future sequences of a time series, MAGNET aims to learn a denoising process under multi-scale guidance conditioned on the past sequence, to reverse a specified diffusion process, in which the future sequence is corrupted to white noise according to a variance schedule. The multi-scale guidance is implemented by deriving representations of the past sequence on different time scales, specifically from local to global scales, through adaptive-sized windows and providing them to denoising step by step. With multi-scale guidance, MAGNET fully considers the hierarchy implied by the scale transition phenomenon and leverage it to guide the denoising process. Also, this guidance contributes to more efficient representation of the past sequence by learning the multi-scale features in a divide-and-conquer way: on one hand, MAGNET only needs to focus on temporal features of a single time scale at each step of denoising, greatly reducing the difficulty of feature learning; on the other hand, the feature learning is well guaranteed to be comprehensive since features on various time scales are included as the window expands continuously until reaching the global size. Furthermore, multi-scale guidance can also be thought of as imposing a type of hierarchical prior constraint to MAGNET's denoising process, which ensures less instability in denoising and consequently reaches higher accuracy. Here, the main contributions of our paper are listed as follows.

- A novel time series forecasting model is proposed, named MAGNET, which provides a diffusion-model-based framework to predict the distribution of a time series' future trajectories based on its past observations.

- A unique mechanism is designed, named multi-scale guidance, to impose an organized restriction on the denoising process. This mechanism enables MAGNET to provide information on varying time scales as guidance to meet the requirements of different denoising stages, facilitates more effective multi-scale feature learning for better historical information representation, and reduces the instability of the entire denoising process, all of which contribute to high-quality forecasting outcomes.

- Extensive experiments are conducted to test the proposed MAGNET, of which the comparison results demonstrate the outperformance of MAGNET over existing approaches under different evaluation metrics, while others provide insight into MAGNET from aspects such as ablation study, sensitivity analysis, etc.

The rest of the paper proceeds as follows. In Section 2, we introduce existing works related to our topic. In Section 3, we formally discuss the details of the proposed MAGNET. In Section 4, we describe the experimental settings and present the experimental results with thorough analysis. Finally, we conclude our work in Section 5.

2. Related Work

2.1. Deep-Learning-Based Time Series Forecasting

Deep-learning-based time series forecasting models can be grouped into two categories, i.e., deterministic models and probabilistic models. Deterministic models predict the specific future values of the time series based on its historical values. Typical works include RNN-based models [15, 16, 17, 18], CNN-based models [19, 14, 20], Transformer-based models [21, 22, 23, 24], etc. In contrast, probabilistic models predict the distribution of the future values of the time series conditioned on past observations. Some works [25, 26] assume the distribution (usually Gaussian) of the time series and apply RNNs or CNNs to estimate its parameter, while others directly utilize deep generative models such as VAEs [27, 28], GANs [29], and diffusion models [30, 31], to generate all possible future trajectories of the time series.

2.2. Diffusion Models

Diffusion models refer to a family of deep generative models, which exhibit state-of-the-art performances compared to traditional generative models like VAEs [8], GANs [9] and flow-based models [32]. Diffusion models are characterized by a diffusion process and a denoising process fixed to two Markov chains, which are used to complete the transformation between original samples and pure noises. Two typical architectures are usually adopted, which are NCSNs (noise-conditioned score networks) [10] and DDPMs (denoising diffusion probabilistic models) [11]. The applications of diffusion models include computer vision [33, 34, 35], graph generation [36, 37, 38], sequence modeling [39, 30, 31], natural language processing [40, 41], etc.

3. MAGNET

In this section, we formally discuss the proposed MAGNET, whose overall architecture is shown in Figure 3. Given a D-variate time series dataset $\mathcal{S} = \{\boldsymbol{S}^i = (\boldsymbol{X}^i, \boldsymbol{Y}^i)|i = 1, 2, ..., N\}$, where $\boldsymbol{X}^i \in \mathbb{R}^{T\times D}$ and $\boldsymbol{Y}^i \in \mathbb{R}^{L\times D}$ denote the T-length past sequence and L-length future sequence of ith sample $\boldsymbol{S}^i$, MAGNET aims to predict the distribution of any $\boldsymbol{Y}^i$ based on the observed $\boldsymbol{X}^i$. As is displayed, MAGNET is with a diffusion-model-based framework, which makes it essentially a conditional distribution transformation between standard Gaussian distribution and data distribution of $\{\boldsymbol{Y}^i|i = 1, 2, ..., N\}$, with $\{\boldsymbol{X}^i|i = 1, 2, ..., N\}$ serving as the condition. Furthermore, MAGNET is especially equipped with a **multi-scale guidance** mechanism to guide the denoising process, aiming to achieve more organized and effective denoising based on hierarchical multi-scale feature utilization. The model details will be introduced in the following subsections. From now on, $\boldsymbol{X}$ and $\boldsymbol{Y}$ will be used to denote past and future sequences, respectively, with superscript i omitted, unless specifically emphasized.

3.1. Overall Framework

Mathematically, forecasting $\boldsymbol{Y}$ based on $\boldsymbol{X}$ is to model the conditional distribution $p(\boldsymbol{Y}|\boldsymbol{X})$. With a diffusion-model-based framework, MAGNET estimates $p(\boldsymbol{Y}|\boldsymbol{X})$ through a learnable Markov chain:

$$p_\theta(\boldsymbol{Y}_0, \boldsymbol{Y}_1, ..., \boldsymbol{Y}_K|\boldsymbol{X}) = p(\boldsymbol{Y}_K)\prod_{k=1}^{K} p_\theta(\boldsymbol{Y}_{k-1}|\boldsymbol{Y}_k, \boldsymbol{X}), \tag{1}$$

$$p(\boldsymbol{Y}_K) = \mathcal{N}(\boldsymbol{Y}_K; \boldsymbol{0}, \boldsymbol{I}), \tag{2}$$

where K denotes the total number of transition steps, $\mathcal{N}(\cdot)$ denotes the Gaussian distribution, and $\boldsymbol{Y}_0 = \boldsymbol{Y}$. Once selecting a form for $p_\theta(\boldsymbol{Y}_{k-1}|\boldsymbol{Y}_k, \boldsymbol{X})$,

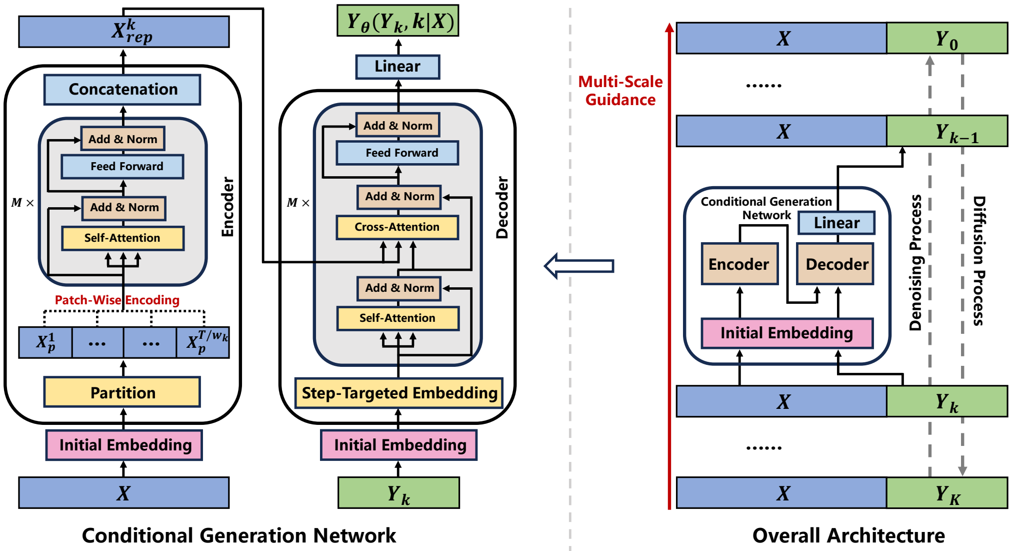


Figure 3: The architecture of the proposed MAGNET.

parameter θ can be learned by reversing another fixed Markov chain that gradually adds noise to $\boldsymbol{Y}$ according to a variance schedule $\{\beta_1, \beta_2, ..., \beta_K\}$:

$$q(\boldsymbol{Y}_1, ..., \boldsymbol{Y}_K|\boldsymbol{Y}_0) = \prod_{k=1}^{K} q(\boldsymbol{Y}_k|\boldsymbol{Y}_{k-1}), \tag{3}$$

$$q(\boldsymbol{Y}_k|\boldsymbol{Y}_{k-1}) = \mathcal{N}(\boldsymbol{Y}_k; \sqrt{1-\beta_k}\boldsymbol{Y}_{k-1}, \beta_k \boldsymbol{I}). \tag{4}$$

In diffusion model parlance, these two Markov chains are referred to as denoising process (reverse process) and diffusion process (forward process), respectively. Obviously, the core component that determines the forecasting performance of MAGNET is the transition function $p_\theta(\boldsymbol{Y}_{k-1}|\boldsymbol{Y}_k, \boldsymbol{X})$. To model it, two questions need to be thought through: *first, how to let* $\boldsymbol{X}$ *provide a condition that guides the whole denoising process to make it more effective in the context of time series forecasting; second, how to accurately model the complex correlations between* $\boldsymbol{X}$ *and* $\boldsymbol{Y}_k$ *for step-wise conditional generation.* To address these two questions, MAGNET introduces a unique **multi-scale guidance**, under which a **conditional generation network** is designed to model $p_\theta(\boldsymbol{Y}_{k-1}|\boldsymbol{Y}_k, \boldsymbol{X})$.

3.2. Multi-Scale Guidance

The multi-scale guidance works by ***making the past sequence self-adjust to initially provide local information, then steadily expand the time scale, and ultimately provide the global information.*** We argue that this guidance is better suited for time series forecasting because of three reasons. First, we hierarchically extract patterns on different time scales from the past sequence and provide them to the denoising process to meet its varying needs at different stages. Second, we introduce a structured scheme to achieve more efficient representation of past observations, where complex multi-scale feature learning is divided into a collection of single-scale feature learning, reducing the difficulty of feature learning at each denoising step while also ensuring the integrity of feature utilization. Third, with such an organized guidance, we impose a prior constraint to denoising process to reduce its instability and make the forecasting more reliable.

3.2.1. Adaptive-Sized Window

The first step to implement the multi-scale guidance is to instantiate a schedule for varying time scales. This is done through an adaptive-sized window. Let w_k denote the window size at denoising step k. It is calculated by

$$w_k = \left\lceil w_{max} - \frac{w_{max} - w_{min}}{K-1}(k-1) \right\rceil, \tag{5}$$

where $\lceil \cdot \rceil$ denotes ceiling operation, w_{min} denotes the minimal window size, and w_{max} denotes the maximal window size, which is set as the look-back length T. In this way, as k ranges from K to 1, we can obtain a window schedule $\{w_K, w_{K-1}, ..., w_1\}$, whose value ranges accordingly from $w_K = w_{min}$ to $w_1 = T$, representing a time scale that expands continuously from local scale to global scale with the denoising process.

3.2.2. Conditional Generation Network

With the window schedule, the next step of the multi-scale guidance is to model the transition function $p_\theta(\boldsymbol{Y}_{k-1}|\boldsymbol{Y}_k, \boldsymbol{X})$. To start with, $p_\theta(\boldsymbol{Y}_{k-1}|\boldsymbol{Y}_k, \boldsymbol{X})$

is specified as

$$p_\theta(\boldsymbol{Y}_{k-1}|\boldsymbol{Y}_k, \boldsymbol{X}) = \mathcal{N}(\boldsymbol{Y}_{k-1}; \boldsymbol{\mu}_\theta(\boldsymbol{Y}_k, k|\boldsymbol{X}), \sigma_k^2 \boldsymbol{I}), \tag{6}$$

$$\begin{aligned}\boldsymbol{\mu}_\theta(\boldsymbol{Y}_k, k|\boldsymbol{X}) &= \frac{\sqrt{\alpha_k}(1-\overline{\alpha}_{k-1})}{1-\overline{\alpha}_k}\boldsymbol{Y}_k \\ &+ \frac{\sqrt{\overline{\alpha}_{k-1}}\beta_k}{1-\overline{\alpha}_k}\boldsymbol{Y}_\theta(\boldsymbol{Y}_k, k|\boldsymbol{X}),\end{aligned} \tag{7}$$

where $\sigma_k^2 = \beta_k$, $\alpha_k = 1 - \beta_k$, $\overline{\alpha}_k = \prod_{i=1}^k \alpha_i$, and $\boldsymbol{Y}_\theta(\cdot)$ denotes a step-wise denoising function. Here, a conditional generation network is introduced to further parameterize $\boldsymbol{Y}_\theta(\cdot)$, which is with a Transformer-type Encoder-Decoder architecture, as is shown in Figure 3.

The role of the Encoder is to represent $\boldsymbol{X}$ at different denoising steps based on the window schedule, to reflect features on different time scales. Before the Encoder, an initial embedding is at first applied to $\boldsymbol{X}$ to map it into $\mathbb{R}^{T\times H}$ space, where H denotes the hidden dimension. Let $\boldsymbol{X}_{in} \in \mathbb{R}^{T\times H}$ denote the input of the Encoder. To derive representation for denoising step k, $\boldsymbol{X}_{in}$ is first evenly partitioned according to window size w_k to generate a patch sequence $\{\boldsymbol{X}_p^i \in \mathbb{R}^{w_k\times D}|i = 1, 2, ..., \lceil T/w_k\rceil\}$. Note that $\boldsymbol{X}_{in}$ will be padded with its last values if T is not divisible by w_k. Then, the self-attention is applied within each patch to update the representation. For instance, $\boldsymbol{X}_p^i$ is updated to $\tilde{\boldsymbol{X}}_p^i$ through M layers of calculations:

$$\tilde{\boldsymbol{X}}_{p,l}^i = F_{enc}(softmax(\frac{\boldsymbol{Q}_{p,l}^i \cdot \boldsymbol{K}_{p,l}^i}{\sqrt{H}})\boldsymbol{V}_{p,l}^i), \forall l \in \{1, 2, ..., M\} \tag{8}$$

$$\boldsymbol{X}_{p,1}^i = \boldsymbol{X}_p^i, \quad \boldsymbol{X}_{p,l+1}^i = \tilde{\boldsymbol{X}}_{p,l}^i, \quad \tilde{\boldsymbol{X}}_p^i = \boldsymbol{X}_{p,M+1}^i, \tag{9}$$

where the input of the lth layer, $\boldsymbol{X}_{p,l}^i$, is updated to $\tilde{\boldsymbol{X}}_{p,l}^i$, $\boldsymbol{Q}_{p,l}^i$, $\boldsymbol{K}_{p,l}^i$ and $\boldsymbol{V}_{p,l}^i$ denote the corresponding query, key and value of $\boldsymbol{X}_{p,l}^i$, and F_{enc} denotes all other calculations such as residual connection, layer normalization, etc., following the paradigm of vanilla Transformer Encoder [42]. All these updated representations are then concatenated to form $\boldsymbol{X}_{rep}^k$, the representation of $\boldsymbol{X}$ for denoising step k that is also the output of the Encoder, by

$$\boldsymbol{X}_{rep}^k = (\tilde{\boldsymbol{X}}_p^1, \tilde{\boldsymbol{X}}_p^2, ..., \tilde{\boldsymbol{X}}_p^{\lceil T/w_k\rceil}). \tag{10}$$

With the derived $\boldsymbol{X}_{rep}^k$, the Decoder subsequently takes it as input for step-wise denoising, from step k to $k-1$. Again, the aforementioned initial embedding is applied to $\boldsymbol{Y}$ before the Decoder to map it into $\mathbb{R}^{T\times H}$ space. Let $\boldsymbol{Y}_{in} \in \mathbb{R}^{T\times H}$ denote the input of the Decoder. To distinguish different denoising steps, $\boldsymbol{Y}_{in}$ is at first updated through a step-targeted embedding to incorporate step k's information, where $\boldsymbol{Y}_{in}$ is added with the Transformer positional embedding [42] of step k. For simplicity of notations, we still use $\boldsymbol{Y}_{in}$ to denote the updated result. Afterwards, an M-layer denoising is applied to $\boldsymbol{Y}_{in}$, each of which consists of a combination of self-attention and cross-attention. Specifically, for any layer l, the input $\boldsymbol{Y}_{in,l}$ is first updated to $\tilde{\boldsymbol{Y}}_{in,l}$ through a self-attention to integrate its own features, and then a cross-attention is calculated between $\tilde{\boldsymbol{Y}}_{in,l}$ and $\boldsymbol{X}^k_{rep}$ to model the correlations between them and extract the information contained in $\boldsymbol{X}^k_{rep}$ to denoise $\tilde{\boldsymbol{Y}}_{in,l}$ to $\boldsymbol{Y}_{out,l}$ that also serves as the input of next layer:

$$\tilde{\boldsymbol{Y}}_{in,l} = F_{dec_1}(softmax(\frac{\boldsymbol{Q}_{Y,l} \cdot \boldsymbol{K}_{Y,l}}{\sqrt{H}})\boldsymbol{V}_{Y,l}), \tag{11}$$

$$\boldsymbol{Y}_{out,l} = F_{dec_2}(softmax(\frac{\boldsymbol{Q}_{\tilde{Y},l} \cdot \boldsymbol{K}_{X,l}}{\sqrt{H}})\boldsymbol{V}_{X,l}), \tag{12}$$

where $\boldsymbol{Q}_{Y,l}$, $\boldsymbol{K}_{Y,l}$ and $\boldsymbol{V}_{Y,l}$ denote the corresponding query, key and value of $\boldsymbol{Y}_{in,l}$, $\boldsymbol{Q}_{\tilde{Y},l}$ denotes the query of $\tilde{\boldsymbol{Y}}_{in,l}$, $\boldsymbol{K}_{X,l}$ and $\boldsymbol{V}_{X,l}$ denote the key and value of $\boldsymbol{X}^k_{rep}$, and F_{dec_1} and F_{dec_2} together denote calculations following the paradigm of vanilla Transformer Decoder [42]. The final output of the Decoder is acquired after M rounds of the above calculation, which is further mapped back to $\mathbb{R}^{T\times D}$ space through a linear layer to serve as the output of $\boldsymbol{Y}_\theta(\cdot)$.

Also, it is worth noting that the initial embedding that we use is of the form

$$InitialEmbed(\cdot) = F_{pe}(\cdot) + F_{ve}(\cdot) + F_{tse}(\cdot), \tag{13}$$

where $F_{pe} : \mathbb{R}^{T\times D} \to \mathbb{R}^{T\times H}$ denotes the positional embedding adopting Transformer positional embedding [42] to embed temporal position information, $F_{ve} : \mathbb{R}^{T\times D} \to \mathbb{R}^{T\times H}$ denotes the value embedding adopting a convolution layer to achieve a shallow feature summarization of $\boldsymbol{X}$, and $F_{tse} : \mathbb{R}^{T\times D} \to \mathbb{R}^{T\times H}$ denotes the time stamp embedding to take into account the frequency of $\boldsymbol{X}$. This technique is suggested and applied by many Transformer-based time series forecasting models and one can refer to [21] for more details.

Algorithm 1 Training

```
repeat:
$k \sim \mathcal{U}(1, 2, .., K)$
$\epsilon \sim \mathcal{N}(0, 1)$
generate $\boldsymbol{Y}_k$ according to 3 and 4
calculate $\boldsymbol{X}_{rep}^k$ according to 8, 9 and 10
calculate $\mathbf{Y}_\theta(\boldsymbol{Y}_k, k|\boldsymbol{X})$ according to 11 and 12
calculate loss $L(\theta)$ according to 14
take gradient step on $\nabla_\theta L(\theta)$
until converged
```

Algorithm 2 Inference

```
$\boldsymbol{Y}_K \sim \mathcal{N}(\boldsymbol{Y}_K; \mathbf{0}, \boldsymbol{I})$
for $k = K, K-1, ..., 1$ do
    $\epsilon \sim \mathcal{N}(0, 1)$ if $k > 1$ else $\epsilon = 0$
    calculate $\boldsymbol{X}_{rep}^k$ according to 8, 9 and 10
    calculate $\mathbf{Y}_\theta(\boldsymbol{Y}_k, k|\boldsymbol{X})$ according to 11 and 12
    calculate $\boldsymbol{\mu}_\theta(\boldsymbol{Y}_k, k|\boldsymbol{X})$ according to 7
    sample $\boldsymbol{Y}_{k-1}$ according to 6
end for
return $\boldsymbol{Y}_0$
```

3.3. Training and Inference

MAGNET is trained to minimize the following loss function:

$$L(\theta) = \mathbb{E}_{\mathbf{Y}_0,k,\epsilon}[||\boldsymbol{Y}_0 - \mathbf{Y}_\theta(\boldsymbol{Y}_k, k|\boldsymbol{X})||^2], \tag{14}$$

where $\epsilon \sim \mathcal{N}(0, 1)$. The training and inference algorithms are presented in Algorithm 1 and Algorithm 2, respectively, where $\mathcal{U}(\cdot)$ denotes uniform distribution.

4. Experiments

4.1. Setup

4.1.1. Datasets

Six datasets are used including both synthetic and real-world data, all of which are split into training, validation and test sets in the ratio 6:2:2. Also, Z-score standardization is applied to preprocess the raw data.

Synthetic Datasets As suggested in [43] and [28], we generate two synthetic datasets, denoted as SD_1 and SD_2, by:

$$\begin{aligned} \boldsymbol{w}_t = a \cdot \boldsymbol{w}_{t-1} + tanh(b \cdot \boldsymbol{w}_{t-2}) \\ + sin(\boldsymbol{w}_{t-3}) + \mathcal{N}(0, 0.5\boldsymbol{I}), \end{aligned} \tag{15}$$

$$\boldsymbol{X} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_T] \cdot \boldsymbol{F} + \mathcal{N}(0, 0.5\boldsymbol{I}), \tag{16}$$

$$0 \leqslant \boldsymbol{w}_{t,1}, \boldsymbol{w}_{t,2} \leqslant 1, \quad t = 1, 2, 3, \tag{17}$$

where $a, b \in \mathbb{R}$, $\boldsymbol{w}_t \in \mathbb{R}^2$, $\boldsymbol{F} \in \mathbb{R}^{2\times n} \sim \mathcal{U}[-1, 1]$. For SD_1, we set $a = 0.7, b = 0.3, n = 10, T = 2000$, while for SD_2, we set $a = 0.5, b = 0.5, n = 20, T = 2000$.

Real-World Datasets Also, we select four real-world datasets. (1) Exchange [14], collecting the daily exchange rates of eight countries including Australia, British, Canada, Switzerland, China, Japan, New Zealand and Singapore from 1990 to 2016. (2) & (3) ETTh1 and ETTh2 [21], containing load and oil temperature data of 2 electricity transformers at 2 stations, recorded hourly from 2016/07 to 2018/07. (4) Weather[1], containing hourly data for 30 meteorological indicators in 2023.

The statistics of these six datasets are summarized in Table 1.

Table 1: Statistics of the six datasets.

Dataset	***D*-variate**	**time steps**	**frequency**
SD_1	10	2,000	1 day
SD_2	20	2,000	1 day
Exchange	8	7,588	1 day
ETTh1	7	17,420	1 hour
ETTh2	7	17,420	1 hour
Weather	30	8,760	1 hour

4.1.2. Evaluation Metrics

Three metrics are selected to measure the performance of MAGNET, which are MAE, MSE and CRPS_{sum}. Among them, MAE and MSE assess the

[1] https://www.bgc-jena.mpg.de/wetter/

accuracy of the mean values of predictions, while CRPS$_{\text{sum}}$ evaluates the accuracy of entire distribution of predictions. Let $\boldsymbol{Y} = (\boldsymbol{Y}_1, \boldsymbol{Y}_2, ..., \boldsymbol{Y}_T) \in \mathbb{R}^{T\times D}$ denote the ground truth of a D-variate time series, and $\hat{\boldsymbol{Y}} = (\hat{\boldsymbol{Y}}_1, \hat{\boldsymbol{Y}}_2, ..., \hat{\boldsymbol{Y}}_T) \in \mathbb{R}^{T\times D}$ denote the corresponding forecast values. The definitions are as follows.

MAE: Mean Absolute Error. The MAE between $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$ is calculated by

$$MAE = \frac{1}{T \times D} \sum_{i=1}^{D} \sum_{j=1}^{T} |\boldsymbol{Y}_{i,j} - \hat{\boldsymbol{Y}}_{i,j}|. \tag{18}$$

MSE: Mean Squared Error. The MSE between $\boldsymbol{Y}$ and $\hat{\boldsymbol{Y}}$ is calculated by

$$MSE = \frac{1}{T \times D} \sum_{i=1}^{D} \sum_{j=1}^{T} |\boldsymbol{Y}_{i,j} - \hat{\boldsymbol{Y}}_{i,j}|^2. \tag{19}$$

CRPS: Continuous Ranked Probability Score. The CRPS is used to evaluate the alignment between the predicted CDF (cumulative distribution function) F and the ground truth observation y, which is defined as

$$CRPS(F, y) = \int_{\mathbb{R}} (F(x) - \mathbb{I}(y \leqslant x))^2 \, dx, \tag{20}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. In the context of time series forecasting, CRPS is typically used for univariate time series or single dimension of a multivariate time series, and is calculated by applying 20 at each time step and then averaging over the entire time horizon. To be more suitable for multivariate time series, CRPS is extended to ***CRPS$_{sum}$*** that is calculated by first applying a summation over the variable dimension of size D to both the predicted and ground truth values, and then comparing the values after summation based on CRPS:

$$CRPS_{sum} = \mathbb{E}_t[CRPS(F_{sum}(t), \boldsymbol{Y}_{sum}(t))], \tag{21}$$

where $F_{sum}(t)$ denotes the CDF of the time series after summation and $\boldsymbol{Y}_{sum}$ denotes the ground truth after summation. The CRPS$_{\text{sum}}$ is the metric used in this paper.

For all these metrics, smaller values indicate better performance.

4.1.3. Baselines

Seven baseline models are selected for comparison as follows.

- *LSTM-MAF* [44]: a conditioned-normalizing-flow-based model combining Masked Autoregressive Flow with LSTM for probabilistic time series forecasting;
- *Transformer-MAF* [44]: a conditioned-normalizing-flow-based model combining Masked Autoregressive Flow with Transformer for probabilistic time series forecasting;
- *TimeGrad* [30]: an RNN conditioned model with a diffusion-model-type framework for probabilistic time series forecasting;
- *D^3VAE* [28]: a VAE-based model with diffusion, denoise and disentanglement for probabilistic time series forecasting;
- *Transformer* [42]: the vanilla Transformer model;
- *Informer* [45]: a Transformer-based model characteristic of *ProbSparse* self-attention, self-attention distilling operation and generative style decoder for deterministic time series forecasting.
- *Autoformer* [46]: a Transformer-based model with inner decomposition and auto-correlation for deterministic time series forecasting.

4.1.4. Implementation

The MAGNET is trained for 100 epochs with early stopping and optimized by Adam [47] with a learning rate 0.0001 and batch size 512. For the diffusion model framework of MAGNET, we set the total number of transition steps $K = 1000$, and select a linear variance schedule for $\{\beta_1, \beta_2, ..., \beta_K\}$, starting with $\beta_1 = 0.0001$ and ending with $\beta_K = 0.02$. To accelerate the sampling, we also adopt the second-order multi-step DPM-Solver++ [48, 49], in which the total number of function evaluations is set as 20. For the conditional generation network of MAGNET, the hidden dimension H is selected from $\{16, 32, 64\}$, while the number of layers M is selected from $\{1, 2\}$. Also, multi-head attention is applied for both Encoder and Decoder, with the same number of heads in each, selected from $\{1, 2, 4\}$. Besides, the reversible instance normalization is also applied, as suggested by [50]. The past sequence length T and future sequence length L are set as $(T = 96, L = 48)$, respectively. For all models, the training-validation-test pipeline is repeated 5 times, and a total of 100 samples are generated for evaluation for all probabilistic models. The experiments relevant to neural network computations are conducted on NVIDIA RTX A4000 16G GPUs.

4.2. Comparison

4.2.1. Quantitative Comparison

Table 2: Comparison between MAGNET and baselines in terms of MAE, MSE and CRPS$_{\text{sum}}$ on six datasets. Note that only MAE and MSE are reported for Transformer, Informer and Autoformer since they are used for deterministic forecasting with CRPS$_{\text{sum}}$ not applicable to them. Each result is reported in the form of mean ± std.

		SD$_1$	**SD$_2$**	**Exchange**	**ETTh1**	**ETTh2**	**Weather**
LSTM-MAF	***MAE***	1.0517 ± 0.1108	0.8314 ± 0.0278	1.8396 ± 0.2051	1.7855 ± 0.0771	2.1063 ± 0.0675	0.8112 ± 0.0319
	MSE	1.8162 ± 0.3628	1.2057 ± 0.1437	6.3278 ± 3.2645	5.3297 ± 0.7354	7.7105 ± 0.4911	1.5483 ± 0.1651
	CRPS$_{sum}$	0.2196 ± 0.0466	0.2915 ± 0.0159	0.3249 ± 0.0362	1.0905 ± 0.0561	0.7065 ± 0.0415	0.4533 ± 0.0221
Transformer-MAF	***MAE***	1.0278 ± 0.3030	0.8203 ± 0.0513	1.7644 ± 0.4416	1.5916 ± 0.0216	1.7433 ± 0.0533	0.8100 ± 0.0621
	MSE	1.7741 ± 0.9905	1.1999 ± 0.2344	7.3289 ± 2.9138	4.6068 ± 0.5487	5.0713 ± 0.4241	1.6939 ± 0.3145
	CRPS$_{sum}$	0.1783 ± 0.0468	0.2802 ± 0.0162	0.2799 ± 0.1022	0.9695 ± 0.0424	0.6984 ± 0.0192	0.4612 ± 0.0789
TimeGrad	***MAE***	1.1660 ± 0.2460	0.7589 ± 0.0347	1.4303 ± 0.1532	0.7721 ± 0.0161	0.7327 ± 0.0517	0.4458 ± 0.0219
	MSE	2.1166 ± 0.8653	0.9725 ± 0.0949	4.1865 ± 1.4192	1.1555 ± 0.0955	1.2204 ± 0.1568	0.5022 ± 0.0565
	CRPS$_{sum}$	0.0478 ± 0.0018	0.0585 ± 0.0030	0.1848 ± 0.0266	0.2214 ± 0.0151	0.1553 ± 0.0221	0.0858 ± 0.0054
D^{3}VAE	***MAE***	1.3783 ± 0.0527	0.9650 ± 0.0383	1.4528 ± 0.0952	0.8786 ± 0.0250	0.9647 ± 0.0810	1.2113 ± 0.1787
	MSE	2.6404 ± 0.1541	1.5102 ± 0.1183	3.4286 ± 0.4006	1.3216 ± 0.0877	1.5416 ± 0.2563	2.5055 ± 0.7658
	CRPS$_{sum}$	0.0737 ± 0.0094	0.0928 ± 0.0064	0.2032 ± 0.0260	0.2417 ± 0.0295	0.2352 ± 0.0279	0.1787 ± 0.0222
Transformer	***MAE***	0.9632 ± 0.0806	0.6302 ± 0.0094	0.6184 ± 0.0483	0.6534 ± 0.0282	0.4063 ± 0.0046	0.4255 ± 0.0136
	MSE	1.3380 ± 0.1885	0.6957 ± 0.0210	0.7132 ± 0.1139	0.7634 ± 0.0452	0.3267 ± 0.0128	0.3624 ± 0.0250
Informer	***MAE***	1.0624 ± 0.0255	0.6369 ± 0.0062	0.9044 ± 0.0222	0.6309 ± 0.0176	0.4707 ± 0.0117	0.4766 ± 0.0182
	MSE	1.6083 ± 0.0662	0.7177 ± 0.0184	1.3884 ± 0.0711	0.7582 ± 0.0123	0.4003 ± 0.0257	0.4377 ± 0.0291
Autoformer	***MAE***	0.7057 ± 0.0052	0.6330 ± 0.0134	0.2853 ± 0.0163	**0.5387 ± 0.0151**	0.3735 ± 0.0135	0.3482 ± 0.0039
	MSE	0.8439 ± 0.0143	0.6890 ± 0.0340	0.1501 ± 0.0193	**0.5853 ± 0.0544**	0.2775 ± 0.0179	0.2828 ± 0.0032
MAGNET	***MAE***	**0.6961 ± 0.0357**	**0.6296 ± 0.0116**	**0.2268 ± 0.0046**	0.6075 ± 0.0186	**0.3611 ± 0.0090**	**0.3091 ± 0.0099**
	MSE	**0.8305 ± 0.0881**	**0.6766 ± 0.0307**	**0.0975 ± 0.0051**	0.7462 ± 0.0634	**0.2700 ± 0.0160**	**0.2681 ± 0.0090**
	CRPS$_{sum}$	**0.0471 ± 0.0022**	**0.0578 ± 0.0065**	**0.0448 ± 0.0020**	**0.2158 ± 0.0130**	**0.1224 ± 0.0063**	**0.0812 ± 0.0012**

The comparison results between MAGNET and baseline models in terms of MAE, MSE and CRPS_{sum} are listed in Table 2. As is shown, MAGNET achieves the best performance in almost all cases. The outperformance of MAGNET over the four probabilistic forecasting models LSTM-MAF, Transformer-MAF, TimeGrad and D^3VAE is significant, whether with regard to MAE and MSE or CRPS_{sum}. This indicates that MAGNET can provide more accurate point estimation for the future values of time series, and also possesses greater distribution characterization capability. In particular, compared with TimeGrad and D^3VAE, the two diffusion-model-based models, MAGNET achieves an average percentage improvement of 40.69% over TimeGrad and 56.09% over D^3VAE on MAE, an average percentage improvement of 58.13% over TimeGrad and 72.70% over D^3VAE on MSE, and an average percentage improvement of 17.91% over TimeGrad and 44.16% over D^3VAE on CRPS_{sum}. This result is quite encouraging as it in a sense suggests that while all these three models are based on the diffusion model architecture, MAGNET is able to achieve the best forecasting outcomes, which may benefit from its more organized denoising process achieved by the multi-scale guidance. In addition, we can also see that except for dataset ETTh1, MAGNET outperforms the three deterministic forecasting models Transformer, Informer and Autoformer on both MAE and MSE. Even on ETTh1, MAGNET is the second best model, only inferior to Autoformer.

4.2.2. Statistical Test

The results discussed above showcase the superiority of MAGNET over baseline models. Here, we further detect whether the superiority holds statistical significance by conducting Friedman test and post-hoc Bonferroni-Dunn test [51]. Briefly, Friedman test is used to examine whether there is significant difference among the performances of a group of models, while Bonferroni-Dunn test is then used to detect whether the target model (MAGNET, in our case) outperforms other models after the Friedman test shows that the difference indeed exists. Both these tests are based on the performance ranks of models, and in our setting, each model is ranked according to its performance in each (dataset, metric) pair. As such, we separately examine the difference between MAGNET and probabilistic models, and the difference from deterministic models, since CRPS_{sum} is not applicable to deterministic models.

The test results are presented in Figure 4, with the significance level of $\alpha = 0.10$. Taking Figure 4 (a) as an example, we can see that the null

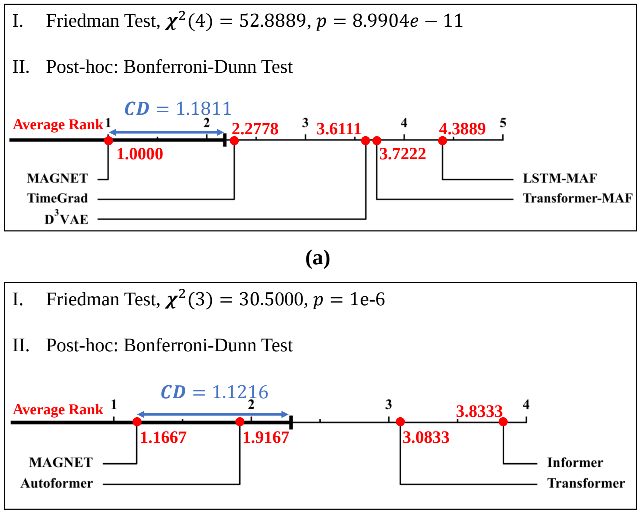


Figure 4: The (a) and (b) correspond to the test results for the differences between MAGNET and probabilistic models, and between MAGNET and deterministic models, respectively.

hypothesis of Friedman test is rejected, indicating that all these five models perform significantly differently. Also, the Bonferroni-Dunn test further suggests that MAGNET outperforms TimeGrad, D^{3}VAE, Transformer-MAF and LSTM-MAF significantly since their distances to MAGNET exceed the CD (critical distance). A similar analysis also applies to the results in Figure 4(b), with the only exception that the distance between MAGNET and Autoformer is within the CD. Even so, the average rank of MAGNET is still smaller than that of Autoformer, just this advantage not statistically significant under current test.

In general, these results further prove the superiority of MAGNET over baselines from a statistical perspective.

4.2.3. Qualitative Comparison

To more intuitively demonstrate the forecasting capability of MAGNET, we visualize MAGNET's forecasting outcomes on datasets ETTh1 and Weather in Figure 5, with only the results on the first three dimensions being displayed. As is shown, in each subfigure, the forecasting mean curve closely

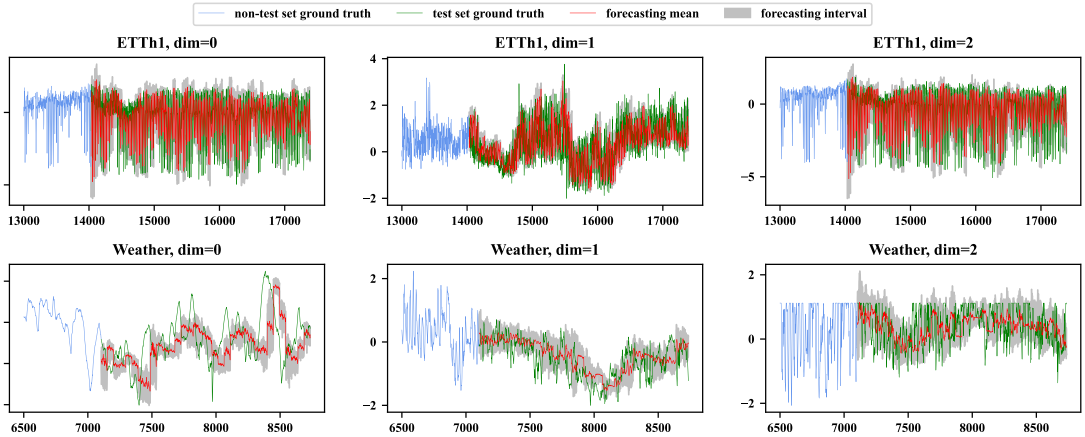


Figure 5: The visualization of MAGNET's forecasting outcomes on ETTh1 and Weather. The forecasting spans the entire test set and a portion of the non-test set data is retained for ease of presentation.

aligns with the test set ground truth curve in shape, which proves the effective point estimation capability of MAGNET from an intuitive visual perspective. Besides, we can also find that the distribution estimation is desirable, with the forecasting interval overall encompassing the test set ground truth curve and with no obvious outliers.

4.3. Analysis

4.3.1. Ablation Study

Recall that the key design in MAGNET is the proposed multi-scale guidance mechanism. Hence, we conduct ablation study here to more directly examine the effectiveness of this mechanism. The experimental results are presented in Table 3. In particular, we denote MAGNET without multi-scale guidance as MAGNET w/o MSG.

As is exhibited, in terms of the mean, MAGNET achieves superior performance over MAGNET w/o MSG measured by all evaluation metrics on all datasets. The greatest improvement with respect to MAE and MSE both occurs on ETTh1, with increases of 10.62% and 24.68%, respectively, while the greatest improvement with respect to CRPS_{sum} occurs on Exchange, with an increase of 17.70%. This demonstrates the indispensability of the proposed

Table 3: Model ablation, comparing the complete MAGNET and MAGNET w/o MSG, in terms of MAE, MSE and CRPS_{sum} on six datasets. Each result is reported in the form of mean ± std.

		SD_1	**SD_2**	**Exchange**	**ETTh1**	**ETTh2**	**Weather**
MAGNET w/o MSG	***MAE***	0.7121 ± 0.0401	0.6386 ± 0.0138	0.2421 ± 0.0053	0.6797 ± 0.0475	0.3795 ± 0.0253	0.3251 ± 0.0112
	MSE	0.8700 ± 0.0982	0.6974 ± 0.0339	0.1117 ± 0.0035	0.9908 ± 0.1762	0.3019 ± 0.0511	0.2950 ± 0.0122
	$\boldsymbol{CRPS_{sum}}$	0.0485 ± 0.0027	0.0587 ± 0.0063	0.0544 ± 0.0033	0.2463 ± 0.0336	0.1302 ± 0.0078	0.0874 ± 0.0029
MAGNET	***MAE***	**0.6961 ± 0.0357**	**0.6296 ± 0.0116**	**0.2268 ± 0.0046**	0.6075 ± 0.0186	**0.3611 ± 0.0090**	**0.3091 ± 0.0099**
	MSE	**0.8305 ± 0.0881**	**0.6766 ± 0.0307**	**0.0975 ± 0.0051**	0.7462 ± 0.0634	**0.2700 ± 0.0160**	**0.2681 ± 0.0090**
	$\boldsymbol{CRPS_{sum}}$	**0.0471 ± 0.0022**	**0.0578 ± 0.0065**	**0.0448 ± 0.0020**	**0.2158 ± 0.0130**	**0.1224 ± 0.0063**	**0.0812 ± 0.0012**

multi-scale guidance in the denoising process to enhance the accuracy of future value generation. Besides, it can also be observed that MAGNET is with an overall smaller standard deviation on these three metrics than MAGNET w/o MSG. This can be taken as a proof of the ability of multi-scale guidance to facilitate a more stable denoising process, which stems from the hierarchical prior constraint it introduces. In general, all these results justify the crucial role that multi-scale guidance plays in achieving effective forecasting for MAGNET.

4.3.2. Sensitivity to Minimal Window Size

The minimal window length w_{min} plays a key role in MAGNET as it controls the hierarchy of the denoising scheme. Note that when w_{min} reaches the length of the past sequence T, MAGNET degenerates to MAGNET w/o MSG, whose performance is discussed before. Here, we work further to explore how the performance of MAGNET changes with the variation of w_{min}. The experimental results are exhibited in Figure 6.

As is shown, it is obvious that the optimal value of w_{min} varies in different situations, with $w_{min} = 6$ for SD_1, $w_{min} = 48$ for SD_2 and Exchange, and $w_{min} = 24$ for the rest. As different datasets have their own exclusive properties, the optimal value of w_{min} is totally dependent on the specific datasets and hard to determine in advance. However, a rough rule can still be summarized that, under most conditions, an excessively large or small value of w_{min} is not recommended since only SD_1 prefers small w_{min} and the performance of MAGNET degrades significantly when w_{min} reaches 96 (the length of the past sequence). This phenomenon can be understood from the view

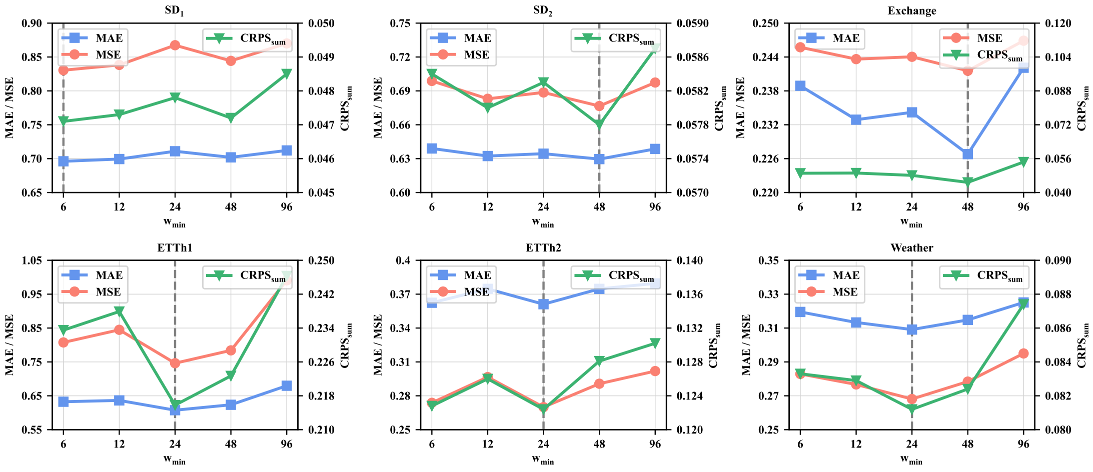
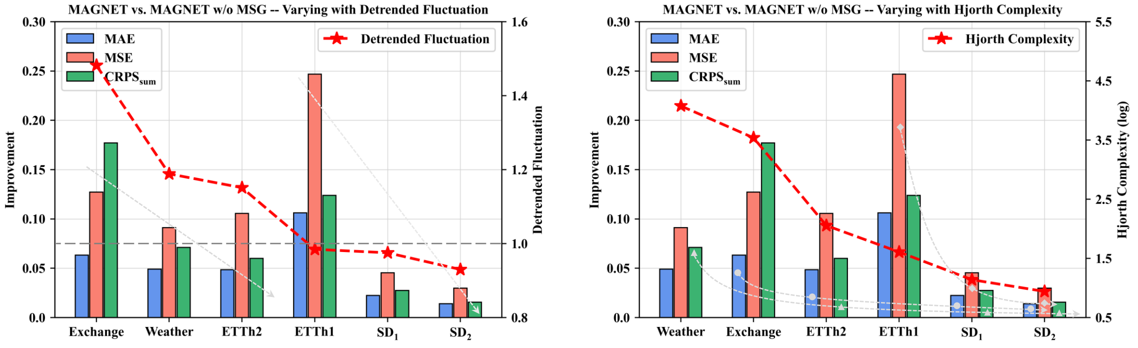


Figure 6: Sensitivity to minimal window size w_{min}.

of feature learning of the past observations, i.e., excessively small w_{min} leads to excessive scale stratification, which may introduce noise, while excessively large w_{min} results in less sufficient local feature learning, both of which can hinder the efficient representation of past information and correspondingly obstruct the forecasting.

4.3.3. Effectiveness under Different Data Complexities

Note that, in order to better represent the past observations to guide the denoising, the proposed multi-scale guidance mechanism facilitates a way of feature learning by hierarchically dividing and integrating the complex multi-scale features of time series, which is therefore expected to be more effective when the time series to be handled possess higher degree of complexity. To investigate this, we apply two complexity indicators, *detrended fluctuation* [52] and *Hjorth complexity* [53] to measure the complexity of our six datasets and track how the outperformance of MAGNET over MAGNET w/o MSG changes with respect to these indicators. Briefly, the detrended fluctuation is often used to determine the self-affinity, especially long-term correlation, of a time series. When the value is less than 1, greater values indicate stronger long-term self-correlation, whereas when the value is greater than 1, higher values indicate stronger non-stationarity. As for Hjorth complexity, it measures the similarity of a time series with a pure sine wave, with higher values indicating greater complexity. The results are reported in Figure 7.



(a)

(b)

Figure 7: Improvement of MAGNET relative to MAGNET w/o MSG under different data complexities, measured by detrended fluctuation and Hjorth complexity.

As is illustrated in Figure 7(a), the threshold 1 splits the value curve of detrended fluctuation into two segments, which also categorizes the six datasets into two groups, where Exchange, Weather and ETTh2, as one group, exhibit more significant non-stationarity, while ETTh1, SD_1 and SD_2, as another group, exhibit more significant long-term self-correlation. At the level above 1, the improvement of MAGNET over MAGNET w/o MSG decreases as the detrended fluctuation of the data decreases across Exchange, Weather and ETTh2. This indicates that MAGNET performs better in environments with stronger non-stationarity with the help of the multi-scale guidance. At the level below 1, it also exhibits a downward trend as the detrended fluctuation of data decreases, which is a strong proof that the multi-scale guidance mechanism takes effect more significantly when data shows stronger long-term self-correlation.

The similar situation also occurs in Figure 7(b), where the improvement on the three metrics still shows a general downward trend as the Hjorth complexity of the data decreases, although there exist exceptions like ETTh1. Considering that Hjorth complexity only measures the complexity of time series from a single viewpoint, it is normal for some datasets to exhibit ”inconsistent” behaviors that may be caused by other factors. Also, if we focus on some specific datasets rather than all of them, this downward trend becomes more significant. For instance, three gray dashed curves are highlighted in Figure 7 to illustrate: the decrease in the improvement on $\mathrm{CRPS}_{\mathrm{sum}}$ as Hjorth complexity decreases across Weather, ETTh2, SD_1 and SD_2; the decrease in the improvement on MAE as Hjorth complexity decreases across Exchange, ETTh2, SD_1 and SD_2; and the decrease in the improvement on MSE as Hjorth complexity decreases across ETTh1, SD_1 and SD_2.

In summary, these results align with our expectations and validate the effectiveness of multi-scale guidance in handling complexity from an experimental perspective.

5. Conclusion

In this paper, we introduce a novel model, MAGNET, for probabilistic time series forecasting. MAGNET possess a diffusion-model-based framework and is specifically equipped with a multi-scale guidance mechanism to impose hierarchical guidance to the denoising. With this guidance, MAGNET is effective in providing guidance on different time scales to meet the demands of varying denoising stages, representing historical information through step-by-step multi-scale feature learning and reducing the instability of denoising outcomes, all of which benefit the forecasting target. MAGNET is tested on six datasets including both synthetic and real-world datasets. The experimental results justify the superior performance of MAGNET over existing baseline models with respect to both point estimation measured by MAE and MSE, as well as distribution prediction measured by $\mathrm{CRPS}_{\mathrm{sum}}$. Other experiments are also conducted for further analysis such as ablation study, sensitivity analysis, and so on.

Future work will mainly concentrate on two aspects to further extend the model. First, some real-world multivariate time series data exist in the form of dynamic graphs, showing more complex variate correlation. In this regard, MAGNET can be further extended to consider this property. A promising way is to combine MAGNET with graph learning methods such as graph neural networks. Also, MAGNET adopts a linear window schedule for multi-scale guidance, which can be further improved to provide more sophisticated guidance, especially when data exhibits stronger complexity.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 62276013, 62141605, 62050132), the Beijing Natural Science Foundation (Grant No. 1192012), and the Fundamental Research Funds for the Central Universities.

References

[1] C. W. J. Granger, P. Newbold, Forecasting economic time series, Academic press, 2014.

[2] M. B. Ntlangu, A. Baghai-Wadji, Modelling network traffic using time series analysis: A review, in: Proceedings of the International Conference on Big Data and Internet of Thing, BDIOT '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 209–215. doi:10.1145/3175684.3175725.
URL https://doi.org/10.1145/3175684.3175725

[3] I. Lana, J. Del Ser, M. Velez, E. I. Vlahogianni, Road traffic forecasting: Recent advances and new challenges, IEEE Intelligent Transportation Systems Magazine 10 (2) (2018) 93–109. doi:10.1109/MITS.2018.2806634.

[4] J. Kaur, K. S. Parmar, S. Singh, Autoregressive models in environmental forecasting time series: a theoretical and application review, Environmental Science and Pollution Research 30 (8) (2023) 19617–19641.

[5] P. A. Adams, T. Adrian, N. Boyarchenko, D. Giannone, Forecasting macroeconomic risks, International Journal of Forecasting 37 (3) (2021) 1173–1191.

[6] J. W. Taylor, K. S. Taylor, Combining probabilistic forecasts of covid-19 mortality in the united states, European Journal of Operational Research 304 (1) (2023) 25–41.

[7] B. Li, J. Zhang, A review on the integration of probabilistic solar forecasting in power systems, Solar Energy 210 (2020) 68–86.

[8] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[10] Y. Song, S. Ermon, Generative modeling by estimating gradients of the data distribution, Advances in neural information processing systems 32 (2019).

[11] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, Advances in neural information processing systems 33 (2020) 6840–6851.

[12] P. Dhariwal, A. Nichol, Diffusion models beat gans on image synthesis, Advances in neural information processing systems 34 (2021) 8780–8794.

[13] J. Ho, T. Salimans, Classifier-free diffusion guidance, arXiv preprint arXiv:2207.12598 (2022).

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

[15] R. Yu, S. Zheng, A. Anandkumar, Y. Yue, Long-term forecasting using higher order tensor rnns, arXiv preprint arXiv:1711.00073 (2017).

[16] Y. G. Cinar, H. Mirisaee, P. Goswami, E. Gaussier, A. Aït-Bachir, V. Strijov, Position-based content attention for time series forecasting with sequence-to-sequence rnns, in: Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14–18, 2017, Proceedings, Part V 24, Springer, 2017, pp. 533–544.

[17] K. Bandara, C. Bergmeir, S. Smyl, Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach, Expert systems with applications 140 (2020) 112896.

[18] S. Smyl, A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting, International Journal of Forecasting 36 (1) (2020) 75–85.

[19] A. Borovykh, S. Bohte, C. W. Oosterlee, Conditional time series forecasting with convolutional neural networks, arXiv preprint arXiv:1703.04691 (2017).

[20] R. Sen, H.-F. Yu, I. S. Dhillon, Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting, Advances in neural information processing systems 32 (2019).

[21] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference, Vol. 35, AAAI Press, 2021, pp. 11106–11115.

[22] Y. Zhang, J. Yan, Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting, in: The Eleventh International Conference on Learning Representations, 2022.

[23] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: International Conference on Machine Learning, PMLR, 2022, pp. 27268–27286.

[24] Y. Nie, N. H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: Long-term forecasting with transformers, in: The Eleventh International Conference on Learning Representations, 2023. URL `https://openreview.net/forum?id=Jbdc0vTOcol`

[25] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, Deepar: Probabilistic forecasting with autoregressive recurrent networks, International Journal of Forecasting 36 (3) (2020) 1181–1191.

[26] Y. Chen, Y. Kang, Y. Chen, Z. Wang, Probabilistic forecasting with temporal convolutional neural network, Neurocomputing 399 (2020) 491–501.

[27] Y. Yuan, K. Kitani, Diverse trajectory forecasting with determinantal point processes, arXiv preprint arXiv:1907.04967 (2019).

[28] Y. Li, X. Lu, Y. Wang, D. Dou, Generative time series forecasting with diffusion, denoise, and disentanglement, Advances in Neural Information Processing Systems 35 (2022) 23009–23022.

[29] A. Koochali, A. Dengel, S. Ahmed, If you like it, gan it—probabilistic multivariate times series forecast with gan, Engineering proceedings 5 (1) (2021) 40.

[30] K. Rasul, C. Seward, I. Schuster, R. Vollgraf, Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting, in: International Conference on Machine Learning, PMLR, 2021, pp. 8857–8868.

[31] L. Shen, J. Kwok, Non-autoregressive conditional diffusion models for time series prediction, arXiv preprint arXiv:2306.05043 (2023).

[32] D. Rezende, S. Mohamed, Variational inference with normalizing flows, in: International conference on machine learning, PMLR, 2015, pp. 1530–1538.

[33] D. Baranchuk, I. Rubachev, A. Voynov, V. Khrulkov, A. Babenko, Label-efficient semantic segmentation with diffusion models, arXiv preprint arXiv:2112.03126 (2021).

[34] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, T. Salimans, Cascaded diffusion models for high fidelity image generation, The Journal of Machine Learning Research 23 (1) (2022) 2249–2281.

[35] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, M. Norouzi, Image super-resolution via iterative refinement, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (4) (2022) 4713–4726.

[36] C. Vignac, I. Krawczuk, A. Siraudin, B. Wang, V. Cevher, P. Frossard, Digress: Discrete denoising diffusion for graph generation, arXiv preprint arXiv:2209.14734 (2022).

[37] L. Kong, J. Cui, H. Sun, Y. Zhuang, B. A. Prakash, C. Zhang, Autoregressive diffusion model for graph generation, in: International Conference on Machine Learning, PMLR, 2023, pp. 17391–17408.

[38] C. Niu, Y. Song, J. Song, S. Zhao, A. Grover, S. Ermon, Permutation invariant graph generation via score-based generative modeling, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 4474–4484.

[39] Z. Kong, W. Ping, J. Huang, K. Zhao, B. Catanzaro, Diffwave: A versatile diffusion model for audio synthesis, arXiv preprint arXiv:2009.09761 (2020).

[40] E. Hoogeboom, D. Nielsen, P. Jaini, P. Forré, M. Welling, Argmax flows and multinomial diffusion: Learning categorical distributions, Advances in Neural Information Processing Systems 34 (2021) 12454–12465.

[41] P. Yu, S. Xie, X. Ma, B. Jia, B. Pang, R. Gao, Y. Zhu, S.-C. Zhu, Y. N. Wu, Latent diffusion energy-based model for interpretable text modeling, arXiv preprint arXiv:2206.05895 (2022).

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[43] A. Farnoosh, B. Azari, S. Ostadabbas, Deep switching auto-regressive factorization: Application to time series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 7394–7403.

[44] K. Rasul, A.-S. Sheikh, I. Schuster, U. Bergmann, R. Vollgraf, Multivariate probabilistic time series forecasting via conditioned normalizing flows, arXiv preprint arXiv:2002.06103 (2020).

[45] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 11106–11115.

[46] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Advances in neural information processing systems 34 (2021) 22419–22430.

[47] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, ArXiv abs/1412.6980 (2014).

[48] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, J. Zhu, Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps, Advances in Neural Information Processing Systems 35 (2022) 5775–5787.

[49] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, J. Zhu, Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, arXiv preprint arXiv:2211.01095 (2022).

[50] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, J. Choo, Reversible instance normalization for accurate time-series forecasting against distribution shift, in: International Conference on Learning Representations, 2021.

[51] J. Demšar, Statistical comparisons of classifiers over multiple data sets, The Journal of Machine learning Research 7 (2006) 1–30.

[52] C.-K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, A. L. Goldberger, Mosaic organization of dna nucleotides, Physical review e 49 (2) (1994) 1685.

[53] B. Hjorth, Eeg analysis based on time domain properties, Electroencephalography and clinical neurophysiology 29 (3) (1970) 306–310.