2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Alleviating Distribution Shift in Time Series Forecasting with an Invertible Neural Network Transformation

Zhiyuan Deng[1,2], Zhe Wu[2✉], Li Su[1,2✉], Yiling Wu[2] and Qingfang Zheng[2]

[1] University of Chinese Academy of Sciences, Beijing, China
[2] Pengcheng Laboratory, Shenzhen, China
dengzhiyuan22@mails.ucas.ac.cn, suli@ucas.ac.cn, {wuzh02, wuyl02, zhengqf01}@pcl.ac.cn

Abstract. The distribution of time series data changes over time, posing challenges for accurate time series forecasting. One common approach to tackle the issue of distribution shift involves transforming the data into a latent space where the impact of the shift is minimized. However, existing methods heavily depend on experienced distribution assumptions and lack guidance on the latent space, leading to sub-optimal performance enhancements. To tackle the above challenges, we propose a new transformation technique to explicitly mitigate the distribution shift between historical and forecast data without any distribution assumptions. Specifically, an Invertible Neural Network Transformation (INNT) is designed to convert data into a smooth latent space. The INNT is constructed to be bidirectional and reversible by a temporal slicing mechanism, thereby preserving all information from the original data. Moreover, the transformation process is guided by a pretraining strategy that aims at reducing distribution divergence within the latent space. Additionally, the proposed method is model-agnostic, allowing for seamless integration into various existing forecasting models. Extensive experiments are conducted to validate the accuracy and generalization of the proposed framework.

Keywords: Multivariate Time Series Forecasting, Data Normalization, Distribution Shift.

1 Introduction

Time series forecasting is essential in various applications, such as traffic, weather, and energy analysis [1-3]. The precise prediction of future trends allows individuals to plan their daily commutes more efficiently. Additionally, governments and managers can make well-informed decisions using advanced forecasting and scheduling techniques. Recently, numerous methods have been developed for time series forecasting[4], including transformer-based networks [5-7] and MLP-based networks [8-10]. A common challenge hindering forecasting advancements is the distribution shift, which means the distribution of time series data changes over time. This shift complicates capturing the relationship between historical and forecast data, leading to limited generalization and suboptimal performance. To address the distribution shift, current studies concentrate

on transforming data into a latent space where historical and forecast data are more proximate. These techniques are primarily implemented through improved normalization methods. For example, RevIN [11], Dish-TS [12], and SAN [13] are essentially variants of conventional normalization to align data with a standard normal distribution. However, these methods exhibit two limitations.

Firstly, the inherent distribution of time series data is notably intricate, and a specific distribution may not adequately capture the complexity of the data. Furthermore, these methods rely on precise future statistics, such as mean and standard deviation. These statistics can be challenging to ascertain due to the nonlinear nature and temporal variability of the data. Despite efforts by existing methods to quantify the shift, the utilization of inaccurate statistics continues to mislead forecasting models. Secondly, these techniques do not impose constraints on the latent space. They rely on a fixed process for data transformation but fail to evaluate its effectiveness. Lacking sufficient guidance and evaluations on transformation results in the failure to effectively reduce the distribution divergence between historical and forecast data.t, either.

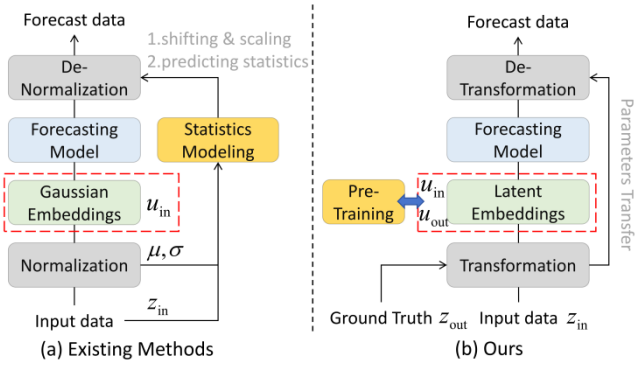

Fig. 1. Comparison with the existing methods, which transform observed data $\mathbf{z}_{\text{in}}$ into latent data $\mathbf{u}_{\text{in}}$. (a) Existing methods rely on future statistics and de-normalize data primarily through shifting and scaling[11] or predicting future statistics[12, 13]. (b) The proposed method transforms input data into latent embeddings and applies a pretraining strategy to minimize the distribution discrepancy between transformed input data $\mathbf{u}_{\text{in}}$ and ground truths $\mathbf{u}_{\text{out}}$.

To tackle the aforementioned challenges, we propose the Invertible Neural Network Transformation (INNT). This approach aims to explicitly minimize distribution divergence between historical and forecast data without any distribution assumptions on time series. As illustrated in Figure 1, compared with the existing methods, INNT offers a bijective mapping for data transformation and de-transformation, and effectively minimizes distribution divergence through transformation pretraining. Firstly, considering the actual distribution is complex and difficult to fit, we design a reversible network as a special distribution transformation to transform historical data into a latent feature space. To ensure the bijection of the transformation, we employ a temporal slicing mechanism to divide a time series into multiple equal-length sub-series. That also meets the scenarios where the temporal dimensions of the input and output series are inconsistent. Secondly, a pretraining strategy is implemented using a slicing maximum mean

discrepancy loss function to optimize the latent space, ensuring that historical and forecast data exhibit similar distributions after transformation. Then, forecasting tasks are carried out within the latent space. Finally, the results are converted back to the observed space through an inverse transformation without relying on future statistics. INNT mitigates distribution shifts by employing a flexible and controllable transformation, thereby enhancing forecasting accuracy and improving model generalization.

The main contributions are concluded as follows:

- We propose the INNT to alleviate the distribution shift problem in time series forecasting without relying on specific distributions, making data transformation more generalized in the scenarios of drastic data changes. % We propose a novel INNT to alleviate the distribution shift problem in time series forecasting without relying on specific distributions or future statistics, making forecasting models more efficient and generalized to unknown distributions.
- We employ a temporal slicing mechanism to satisfy the bijection of data transformation, avoiding information loss.
- We employ pretraining to guide data transformation, explicitly reducing distribution divergence between historical and forecast data in the latent space.
- Extensive experiments demonstrate the superiority of our method in both accuracy and generalization.

2 Related Works

2.1 Time Series Forecasting

Time series forecasting has become a research hotspot, achieving significant advancements in recent years. In the early stage, statistical methods like Average Regressive Integrated Moving Average (ARIMA) and Historical Average (HA) are applied to construct the distribution and prediction of data. With the rapid development of deep learning architectures, time series forecasting has been greatly promoted. For example, Transformer[14] has been widely adopted in time series forecasting. Numerous studies, including CrossFormer [6], InFormer [14], AutoFormer [5], FedFormer [7], and iTransformer [15], have enhanced the attention mechanism to better suit time series tasks. Recently, MLP-based methods like DLinear [8], TiDE [9], FreTS [16], and FITS [17] have demonstrated superior performance in both accuracy and efficiency. Additionally, TimesNet [18] applies Convolution Neural Network (CNN) and WTRAN [19] applies Recurrent Neural Network (RNN) to extract the periodicity of time series data.

2.2 Time Series Forecasting

Most of the existing approaches focus on exploring superior architectures for time series forecasting, but rarely pay attention to the distribution shift problem. AdaRNN [20] alleviates the impact of nonstationary factors by distribution characterization and matching in RNNs. Koopa [21] transforms data into a latent feature space to mitigate distribution shift, but it only considers the distribution divergence within input windows, neglecting the divergence between input and output windows. Nonstationary

Transformer [22] improves the attention mechanism by incorporating nonstationary factors. However, these methods are tailored to specific models, which may limit their applicability to newer architectures. Normalization-based methods have been effective in mitigating distribution shift. RevIN [11] proposes a simple yet effective normalization-and-denormalization process to mitigate distribution divergence by trainable scaling and shifting. Dish-TS [12] considers the distribution shift in intra-space and inter-space and quantifies the shift by knowledge-induced training. SAN [13] dynamically predicts the distributions of future slices for nonstationary time series forecasting. However, they suffer from specific assumptions on the distribution and reliance on unknown future statistics. To enhance the generalization, In-Flows [23] uses an invertible network to transform data into a latent space. However, it lacks sufficient guidance on transformation, which hinders the effective reduction of distribution divergence between historical and forecast data.

3 Problem Definition

3.1 Time Series Forecasting

Time series forecasting aims to predict future data with input historical data. Take input data as $\{\mathbf{X}^{(t-L)}, \dots, \mathbf{X}^{(t-1)}\} \in \mathbb{R}^{L\times N\times K}$ and forecast data as $\{\widehat{\mathbf{X}}^{(t)}, \widehat{\mathbf{X}}^{(t+1)} \dots, \widehat{\mathbf{X}}^{(t+H-1)}\} \in \mathbb{R}^{H\times N\times K}$. L and H are the lengths of the input and output series, respectively. N and K denote the number of features and feature dimension. The task can be formulated as:

$$\{\widehat{\mathbf{X}}^{(i)}\}_{i=t}^{t+H-1} = \mathscr{F}_\theta\left(\{\mathbf{X}^{(i)}\}_{i=t-L}^{t-1}\right) \tag{1}$$

where θ denotes parameters of the forecasting model $\mathscr{F}$.

3.2 Distribution Shift in Time Series Forecasting

To alleviate the distribution shift, a general paradigm is to use a transformation block $\mathcal{T}$ to transform the input data before forecasting, and a de-transformation block $\mathcal{T}^{-1}$ to map the observed distribution after forecasting. This paradigm helps mitigate the distribution shift between input and output space in forecasting models. Existing works generally define $\mathcal{T}$ as the variants of standard normalization and $\mathcal{T}^{-1}$ as the corresponding inverse operation. A general paradigm of time series forecasting with transformation and de-transformation can be formulated as follows:

$$\{\widehat{\mathbf{X}}^{(i)}\}_{i=t}^{t+H-1} = \mathcal{T}^{-1}(\mathscr{F}_\theta(\mathcal{T}(\{\mathbf{X}^{(i)}\}_{i=t-L}^{t-1}))) \tag{2}$$

4 Methodology

4.1 Overview

In this paper, we propose the Invertible Neural Network Transformation (INNT), a model-agnostic technique involving transformation before forecasting and de-transfor-

mation after forecasting to mitigate distribution shift. Unlike conventional normalization-based approaches, INNT does not rely on prior knowledge of data distribution or precise future statistics, making it more adaptable and less susceptible to errors from inaccurate future information. Additionally, INNT incorporates a pretraining strategy to explicitly minimize distribution divergence between historical and forecast data.

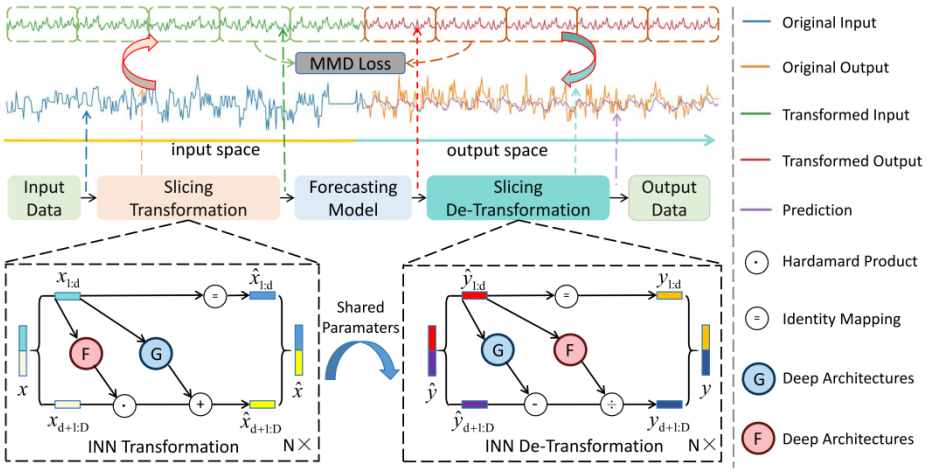

Fig. 2. Overview of INNT. INNT provides a mode-agnostic paradigm addressing distribution shift, including transformation before forecasting and de-transformation afterward. Firstly, a transformation block is used to transform the original input data into a smooth latent space. Then, forecasting tasks are carried out based on the transformed input. Finally, forecasting results are de-transformed to map the observed distribution.

As illustrated in Figure 2, INNT establishes a one-to-one mapping between the observed space and latent space. Initially, a reversible network is utilized to convert the input data into the latent space. Subsequently, the transformed series is fed into a forecasting model $\mathcal{F}_\theta$. The forecasting process is carried out within the transformed latent space. Finally, the output sequence of the forecasting model is transformed back to align with the observed distribution through the corresponding de-transformation process. Importantly, transformation and de-transformation can be trained by minimizing the discrepancy between the historical and forecast series in the latent space, effectively mitigating distribution shift.

4.2 Slicing Transformation and De-Transformation

We propose an invertible transformation $\mathcal{T}_\phi: \mathcal{Z} \to \mathcal{U}$ to convert data from the observed space $\mathcal{Z}$ to a smooth latent space $\mathcal{U}$. The proposed INN transformation $\mathcal{T}_\phi$ and its inverse $\mathcal{T}_\phi^{-1}$ can be defined as:

$$\mathbf{u} \sim p_{\mathcal{U}}(\mathbf{u}), \mathbf{z} \sim p_{\mathcal{Z}}(\mathbf{z}), \mathbf{u} = \mathcal{T}_\phi(\mathbf{z}), \mathbf{z} = \mathcal{T}_\phi^{-1}(\mathbf{u}) \tag{3}$$

where $p_{\mathcal{U}}$ and $p_{\mathcal{Z}}$ denote the latent distribution without specific assumptions and the observed distribution, respectively. $\mathbf{z}$ and $\mathbf{u}$ denote the observed and latent data embeddings.

A critical challenge is to design a transformation module that satisfies the following requirements. First, transformation should be equivalent and reversible, without causing any information loss. Second, any experienced assumption on distribution will limit models to deal with uncertain distributions, so it should not be generalized to one specific distribution.

Inspired by the above requirements, we define the special transformation $\mathcal{T}_\phi$ and de-transformation $\mathcal{T}_\phi^{-1}$ with an Invertible Neural Network (INN) [25] which can reversibly transform data between observed space and latent space.

Initially, ensuring the reversibility of the transformation necessitates consistency between the input and output series in terms of series lengths, which is often challenging to fulfill in practical applications. For instance, forecasting a 336-length future series based on a 96-length historical series may pose difficulties. One approach [24] to address this issue is to transform data on the unchanging feature dimension. However, focusing on the transformation of feature dimension is rarely beneficial to solving the distribution shift in the time dimension. To address the above issue, we apply a temporal slicing mechanism to ensure consistency on the time dimension in INNT. We first slice the series into equal-length periodic segments, denoted as $\mathcal{S}: \mathbb{R}^{N*T} \to \mathbb{R}^{N*T'*P}$, where N is the number of features and T is the length of time series. The temporal slicing operation transforms a T-length series into T' sub-series with a period length P. For example, an hourly collected dataset will be divided into multiple 24-length sub-series to capture the daily periodicity.

Then, we apply INNT on the input sub-sequence $\mathbf{X} \in \mathbb{R}^{N*T'}$ into two parts as $\{\mathbf{X}_{1:d}, \mathbf{X}_{d+1:D}\}$. Transformation can be defined as:

$$\widehat{\mathbf{X}}_{1:d} = \mathbf{X}_{1:d} \tag{4}$$

$$\widehat{\mathbf{X}}_{d+1:D} = \mathbf{X}_{d+1:D} \odot \exp(F(\mathbf{X}_{d+1:D})) + G(\mathbf{X}_{1:d}) \tag{5}$$

where d is the segmentation position, D is the length of series, and $\widehat{\mathbf{X}} = concate(\widehat{\mathbf{X}}_{1:d}, \widehat{\mathbf{Y}}_{d:D})$ is the transformed results. $F, G: \mathbb{R}^d \to \mathbb{R}^{D-d}$ denote deep learning architectures for scaling and shifting. We incorporate MLP layers and tanh activation functions as F and G operations.

Forecasting tasks are performed in a transformed latent space by any forecasting model $\mathscr{F}_\theta$. Take $\widehat{\mathbf{Y}}$ as the forecasting results. The corresponding de-transformation can be defined as:

$$\mathbf{Y}_{1:d} = \widehat{\mathbf{Y}}_{1:d} \tag{6}$$

$$\mathbf{Y}_{d+1:D} = [\widehat{\mathbf{Y}}_{d+1:D} - G(\widehat{\mathbf{Y}}_{1:d})] \odot \exp(F(\widehat{\mathbf{Y}}_{1:d})^{-1}) \tag{7}$$

where $\mathbf{Y} = concate(\mathbf{Y}_{1:d}, \mathbf{Y}_{d:D})$ is the final prediction. Transformation and de-transformation share the same parameters ϕ to maintain a consistent latent space.

One transformation only changes half of the series, so we execute multiple transformations with self-exchange as $\widehat{\mathbf{X}}_{1:d}, \widehat{\mathbf{X}}_{d:D} = \widehat{\mathbf{X}}_{d+1:D}, \widehat{\mathbf{X}}_{1:d}$ to ensure adequate information mixture.

The effective invertible transformation converts data into a smooth latent space, enabling forecasting models to more accurately infer the relationships between historical and forecast series. Parameters are shared between the transformation and de-transformation processes, ensuring that the INNT maintains the equivalence between the observed space and the latent space. This approach effectively enhances time series forecasting from a stable perspective.

4.3 Two-Stage Training Strategy

Though the designed invertible transformation offers a flexible mapping approach without specifying data distribution, a significant challenge lies in effectively training the parameters of the transformation and forecasting models. Specifically, it is necessary to optimize the transformation parameters ϕ to minimize the discrepancy between historical and forecast sequences in the latent space, and the forecasting parameters θ to minimize the error between the predicted sequence and the ground truth. To achieve these goals, the training process is divided into two stages: transformation pretraining and forecasting training.

Transformation Pretraining A slicing Maximum Mean Divergence (MMD) loss function is proposed to evaluate the discrepancy between the transformed historical and forecast series in the latent space.

Transformation is applied on both historical and forecast sub-series as $x^{'} = \mathcal{T}(x) and y^{'} = \mathcal{T}(y)$. $\{x, x^{'}, y, y^{'}\}$ are period-length sub-series. We use slicing MMD loss to evaluate the divergence between input and output sub-series as:

$$loss_{\text{MMD}} = \sum_{k=1}^{N} \backslash | \frac{1}{n^2} \sum_{i,j=1}^{n} f\left(x_{i,k}^{'}, x_{j,k}^{'}\right) - \frac{2}{nm} \sum_{i,j=1}^{n,m} f\left(x_{i,k}^{'}, y_{j,k}^{'}\right) + \frac{1}{m^2} \sum_{i,j=1}^{n,m} f\left(y_{i,k}^{'}, y_{j,k}^{'}\right) \tag{8}$$

where n and m are the numbers of historical and forecast sub-series. The total loss is evaluated across the N features. The kernel function $f(.)$ is defined as $f(a,b) = e^{-\frac{|a-b|^2}{2\sigma^2}}$.

To ensure consistency between the latent space and observed space, we maximize the likelihood of the marginal of the latent embeddings $\mathbf{u}$ via Equation (9) in transformation training, which follows [26].

$$loss_{\mathcal{T}} = \log \left| \det \frac{\partial \mathcal{T}_{\phi}^{-1}(\mathbf{u})}{\partial \mathbf{u}} \right| \tag{9}$$

where $\frac{\partial \mathcal{T}_{\phi}^{-1}(\mathbf{u})}{\partial \mathbf{u}}$ denotes the Jacobian of $\mathcal{T}_{\phi}^{-1}(\mathbf{u})$ at $\mathbf{u}$ and $\det(.)$ computes the determinant.

Ultimately, we learn an invertible mapping function that transforms a time series into a latent representation without information loss. The final loss function of pretraining is:

$$loss_{\text{trans}} = loss_{\mathcal{T}} + loss_{\text{MMD}} \tag{10}$$

In transformation training, only transformation parameters ϕ are considered, and we focus on creating an invertible mapping to a latent space that is equivalent to the observed space but with fewer distribution shifts.

Forecasting Training For the second stage, the goal is to enhance the accuracy of time series forecasting. Since the transformation is essential for the final prediction, both forecasting parameters θ and transformation parameters ϕ are optimized in this stage. The optimization of θ and ϕ is defined as:

$$\theta^*, \phi^* = arg \min_{\phi,\theta} loss_{\text{MSE}} \left(\theta, \phi, \left(\boldsymbol{y}_i^{truth}, \boldsymbol{y}_i^{pred}\right)\right) \tag{11}$$

Overall, the main architecture of INNT is model-agnostic and follows a general paradigm of transformation and de-transformation. The proposed invertible mapping can improve the generalization and accuracy of dealing with drastic distribution changes in time series.

5 Experiments

5.1 Experimental Setup

Datasets We conduct experiments on seven real-world time series datasets : (i) **ETT-datasets**, including ETTh1, ETTh2, ETTm1, and ETTm2. The ETTh datasets are collected hourly, and the ETTm datasets are collected at 15-minute intervals, each containing seven variables. (ii) **Electricity** dataset records the hourly electricity consumption for 321 clients. (iii) **Weather** dataset records the 10-minute evolution of 21 meteorological variables. (iv) **Traffic** dataset, collected from the PEMS system, records hourly traffic density changes from 861 highway sensors.

Implementation All experiments are conducted using PyTorch on two 24GB RTX 4090 GPUs. Input windows are fixed as 96 for PatchTST, 96 for iTransformer, and 336 for DLinear to satisfy the basic settings of models. Horizon windows are set to {96, 336, 720} to evaluate the applicability across different tasks.

Metrics We evaluate time series forecasting performance using Mean Squared Error (MSE) and Mean Absolute Error (MAE). Considering the heterogeneity and varying scales of features, evaluation metrics are normalized for consistency.

Baselines INNT is model-agnostic and can be easily integrated with any forecasting model. We demonstrate the effectiveness of INNT from two aspects. First, INNT enhances the accuracy of existing forecasting methods. We apply INNT to DLinear [8], PatchTST [26], and iTransformer [15] as a plugin. Second, INNT outperforms other methods designed to address the distribution shift problem, including RevIN [11], Dish-TS [12], and SAN [13].

Table 1. Main Results on Different Datasets with Different Forecasting Models.

Method		DLinear		W/ INNT		PatchTST		W/ INNT		iTransformer		W/ INNT	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.372	0.395	0.370	0.392	0.422	0.435	0.411	0.421	0.393	0.410	0.382	0.400
	336	0.437	0.439	0.432	0.430	0.461	0.456	0.420	0.431	0.485	0.461	0.479	0.458
	720	0.557	0.551	0.442	0.456	0.455	0.508	0.438	0.480	0.499	0.487	0.485	0.478
	avg	0.455	0.462	0.415	0.426	0.446	0.466	0.423	0.444	0.459	0.453	0.449	0.445
ETTh2	96	0.295	0.358	0.274	0.336	0.327	0.369	0.305	0.350	0.337	0.380	0.303	0.347
	336	0.473	0.475	0.371	0.407	0.451	0.466	0.423	0.432	0.456	0.454	0.419	0.429
	720	0.729	0.607	0.395	0.430	0.472	0.475	0.423	0.440	0.451	0.461	0.427	0.444
	avg	0.499	0.480	0.347	0.391	0.417	0.437	0.384	0.407	0.415	0.432	0.383	0.407
ETTm1	96	0.300	0.344	0.301	0.346	0.368	0.398	0.350	0.381	0.348	0.376	0.326	0.364
	336	0.372	0.389	0.370	0.386	0.451	0.433	0.416	0.412	0.437	0.425	0.419	0.413
	720	0.424	0.421	0.425	0.417	0.429	0.437	0.398	0.401	0.510	0.463	0.480	0.450
	avg	0.365	0.385	0.365	0.383	0.416	0.423	0.388	0.398	0.432	0.421	0.408	0.409
ETTm2	96	0.169	0.264	0.165	0.255	0.182	0.268	0.186	0.271	0.209	0.291	0.186	0.261
	336	0.292	0.351	0.277	0.329	0.311	0.351	0.310	0.347	0.325	0.358	0.309	0.339
	720	0.443	0.452	0.367	0.384	0.411	0.402	0.411	0.404	0.428	0.425	0.412	0.401
	avg	0.301	0.356	0.270	0.323	0.301	0.340	0.302	0.341	0.321	0.358	0.302	0.334
ELC	96	0.140	0.237	0.140	0.235	0.231	0.320	0.228	0.315	0.152	0.244	0.165	0.256
	336	0.169	0.267	0.171	0.264	0.248	0.336	0.242	0.331	0.182	0.274	0.185	0.275
	720	0.203	0.301	0.209	0.295	0.261	0.355	0.256	0.350	0.218	0.305	0.297	0.314
	avg	0.171	0.268	0.173	0.265	0.247	0.337	0.242	0.332	0.184	0.274	0.190	0.279
Weather	96	0.175	0.237	0.177	0.227	0.208	0.246	0.202	0.244	0.209	0.250	0.211	0.256
	336	0.261	0.312	0.267	0.296	0.264	0.304	0.252	0.298	0.303	0.319	0.306	0.322
	720	0.324	0.36	0.334	0.342	0.318	0.359	0.315	0.352	0.372	0.362	0.373	0.365
	avg	0.253	0.304	0.259	0.288	0.263	0.303	0.256	0.298	0.295	0.310	0.297	0.314
Traffic	96	0.410	0.282	0.409	0.279	0.660	0.445	0.640	0.419	0.414	0.285	0.417	0.286
	336	0.435	0.295	0.435	0.289	0.656	0.442	0.636	0.414	0.450	0.298	0.446	0.289
	720	0.465	0.314	0.462	0.306	0.371	0.481	0.643	0.452	0.481	0.316	0.480	0.315
	avg	0.437	0.297	0.435	0.291	0.662	0.456	0.640	0.428	0.448	0.301	0.448	0.297

5.2 Main Results

In this section, we evaluate the impact of INNT on accuracy by integrating it with various forecasting models, including DLinear [8], PatchTST [15], and iTransformer [15]. Table 1 shows how INNT enhances the performance of three state-of-the-art models across seven datasets.

We make the following observations: (1) Integrating INNT with the original models, which typically overlook distribution shifts, significantly reduces errors on both MSE and MAE metrics. (2) The improvement is especially pronounced for unstable datasets like **ETTH**, which means addressing shift information is essential for irregular and uncertain distribution. (3) For datasets **Electricity** and **Weather**, which have significant

feature scaling differences, iTransformer, which is sensitive to multivariate correlations, exhibits less benefit from INNT. However, INNT still achieves improvements in most settings.

5.3 Comparison with Normalization-based Methods

In this section, we compare the performance of INNT with normalization-based methods, as shown in Table 2 and Table 3. For clarity, we present the average metrics across prediction horizons [96, 336, 720] on DLinear [8] and iTransformer [16]. Baselines include state-of-the-art normalization methods such as SAN [13], Dish-TS [12], and RevIN [11], all of which address distribution shift in time series forecasting. Unlike these baselines, INNT does not rely on specific distribution assumptions or future statistics. These comparisons demonstrate the effectiveness of INNT.

We make the following observations: (1) INNT outperforms these baselines on most datasets and metrics, demonstrating the effectiveness of modeling transformations without relying on specific distributions or accurate future statistics. This improvement is largely due to the more generalized latent space, which better captures data patterns and mitigates distribution shift. (2) Dish-TS [12] underperforms because it is designed for raw time series data, whereas most methods, including the baselines, utilize data pre-processing steps such as standard normalization before training. (3) We conducted comparisons across different forecasting models (DLinear and PatchTST), confirming the universality of the conclusions.

Table 2. Comparison with Normalization-based Methods on DLinear

Methods	DLinear w/ INNT		DLinear w/ SAN		DLinear w/ Dish-TS		DLinear w/ RevIN		Improved	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.415	0.426	0.439	0.440	0.848	0.707	0.416	0.430	0.9%	0.7%
ETTh2	0.347	0.391	0.348	0.392	3.066	1.342	0.374	0.413	5.3%	3.5%
ETTm1	0.365	0.383	0.357	0.386	0.887	0.711	0.382	0.390	-0.7%	2.0%
ETTm2	0.270	0.323	0.264	0.324	3.047	1.348	0.318	0.353	-2.3%	3.8%
Electricity	0.173	0.265	0.190	0.279	0.371	0.428	0.185	0.277	1.7%	1.0%
Weather	0.259	0.288	0.236	0.282	0.648	0.619	0.298	0.322	-0.8%	-0.6%
Traffic	0.435	0.291	0.448	0.304	0.695	0.387	0.422	0.298	6.2%	1.0%

5.4 Ablations

To illustrate how INNT influences the forecasting models, we conduct an ablation study using DLinear and iTransformer. We design three variants as **INNT** (the original model), **w/ INNT** (the proposed method), and **w/ INNT w/o pretrain** (the proposed method without pretraining). As shown in Table 4 and Table 5, **w/ INNT** achieves lower prediction errors compared to the original models **w/o INNT**, confirming that INNT enhances forecasting by effectively reducing distribution shift.

Notably, **w/ INNT w/o pretrain** also shows improvements over the original models in some datasets, suggesting that even without pre-training, the introduction of INNT

Table 3. Comparison with Normalization-based Methods on PatchTST

Methods	PatchTST w/ INNT		PatchTST w/ SAN		PatchTST w/ Dish-TS		PatchTST w/ RevIN		Improved	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.423	0.444	0.427	0.447	0.978	0.679	0.473	0.460	5.6%	3.2%
ETTh2	0.384	0.407	0.405	0.422	2.903	1.309	0.403	0.420	0.3%	0.3%
ETTm1	0.388	0.398	0.391	0.406	0.856	0.713	0.441	0.427	2.2%	0.8%
ETTm2	0.302	0.341	0.309	0.354	3.018	1.005	0.314	0.345	2.1%	0.4%
Electricity	0.242	0.332	0.238	0.329	0.322	0.501	0.254	0.337	-8.9%	-5.3%
Weather	0.256	0.298	0.258	0.300	0.574	0.565	0.396	0.312	9.8%	2.1%
Traffic	0.640	0.428	0.682	0.433	0.702	0.485	0.671	0.433	2.8%	4.3%

can still enhance forecasting performance. This is likely because the forecasting training process helps guide the latent space to be more beneficial for the tasks. Furthermore, the comparison between **w/ INNT** and **w/ INNT w/o pretrain** indicates that pretraining with a slicing MMD loss function can reduce uncertainty and improve accuracy in forecasting, providing a more stable transformation.

Table 4. Ablation Study on DLinear

Methods	DLinear w/ INNT		DLinear w/o INNT		PatchTST w/o INNT w/o pretrain	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.415	0.426	0.455	0.462	0.407	0.424
ETTh2	0.374	0.391	0.499	0.480	0.349	0.396
ETTm1	0.365	0.383	0.365	0.385	0.368	0.383
ETTm2	0.270	0.323	0.301	0.356	0.322	0.356
Electricity	0.173	0.265	0.171	0.268	0.177	0.269
Weather	0.259	0.288	0.253	0.304	0.281	0.307
Traffic	0.435	0.291	0.437	0.297	0.439	0.294

Table 5. Ablation Study on iTransformer

Methods	DLinear w/ INNT		DLinear w/o INNT		PatchTST w/ INNT w/o pretrain	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.449	0.445	0.459	0.453	0.457	0.451
ETTh2	0.383	0.407	0.415	0.432	0.409	0.427
ETTm1	0.408	0.409	0.432	0.421	0.428	0.420
ETTm2	0.302	0.334	0.321	0.358	0.322	0.356
Electricity	0.190	0.279	0.184	0.274	0.184	0.275
Weather	0.297	0.314	0.295	0.310	0.295	0.312
Traffic	0.448	0.297	0.448	0.300	0.447	0.299

5.5 Visualization Analysis

Visualizations on Forecasting Results In this section, we visualize the original and transformed data of **DLinear w/ INNT** on the ETTh2 and ETTm1 datasets, as shown in Figure 3. We make the following observations: (1) The observed series shows irregular and drastic changes, while, in the latent space, the divergence between historical and forecast series is substantially reduced. (2) INNT effectively bridges two distributions with significant differences. Since INNT is a bijective mapping, latent embeddings can be converted back to the observed space without information loss.

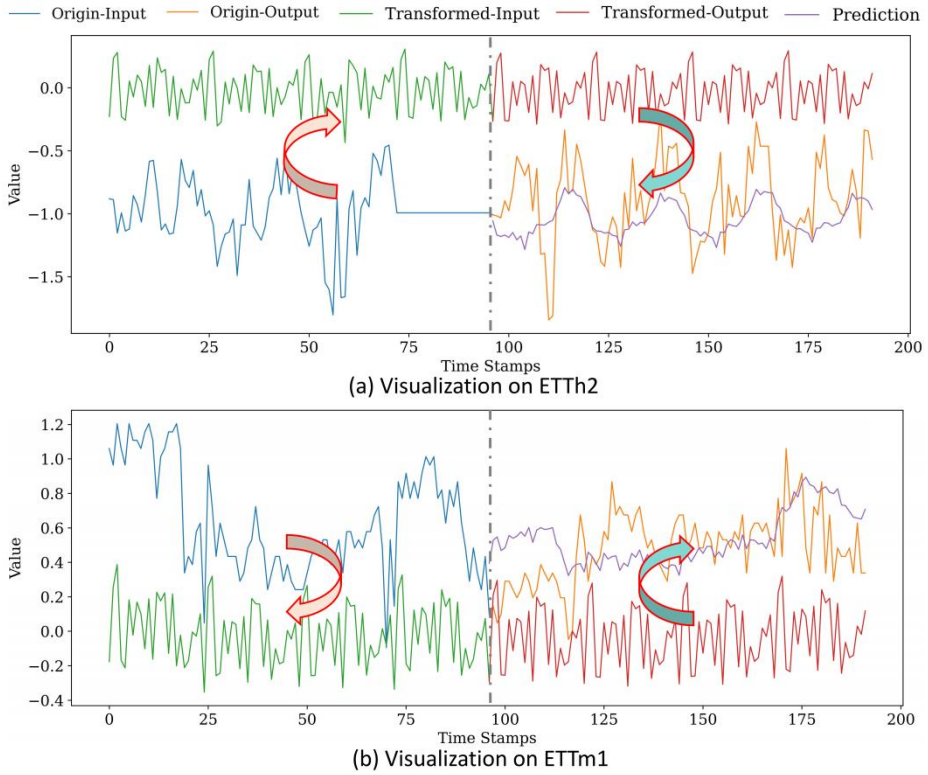

Fig. 3. Visualization of the prediction results of **DLinear w/ INNT** on the ETTh2 and ETTm1 datasets.

Visualizations on the Data Embeddings Normalization-based methods rely on accurate future statistics, such as mean and standard deviation, which are difficult to obtain due to the nonlinearity and temporal instability of time series data. Although methods like RevIN [11], Dish-TS [12], and SAN [13] attempt to quantify these shifts, inaccurate statistics often hinder reducing the distribution shift between input and output spaces. As shown in Figure 4, input and output embeddings after SAN normalization [13] ex-

hibit clear stratification in distribution. In contrast, the proposed method can better alleviate distribution shifts between input and output embeddings in the latent space. Meanwhile, the MMD values between the input and output series are significantly reduced. This demonstrates the effectiveness of the INNT to reduce distribution divergence in transformation.

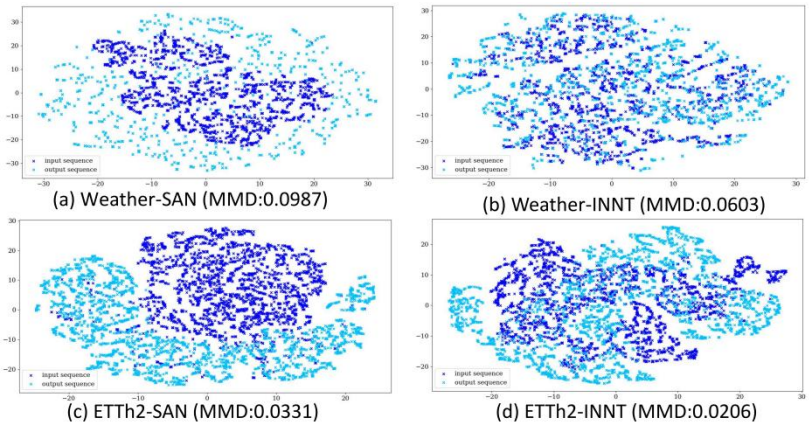

Fig. 4. Visualization of the data embeddings from the normalization method (SAN [13]) and the proposed method by T-SNE [27]. Deep blue points represent the transformed input embeddings and light blue points represent the transformed output embeddings.

6 Conclusion

This paper aims to alleviate the distribution shift problem in time series forecasting through a generalized transformation approach. The accuracy and generalization capabilities of current methods are often limited by specific distribution assumptions and a lack of sufficient guidance for transformation procedures. To tackle these limitations, we present a model-agnostic framework named INNT, which transforms data into a smooth latent space with a bijective mapping. Without reliance on predefined distribution assumptions or unattainable future statistics, INNT is better suited to handle uncertain distributions and provide more precise predictions. Additionally, we explicitly minimize the distribution discrepancy between historical and forecast data through a pretraining strategy on transformation. Extensive experiments have been conducted to validate the effectiveness of the proposed framework.

References

1. X. Han, X. Zhang, Y. Wu, Z. Zhang, T. Zhang, and Y. Wang: Knowledge-based multiple relations modeling for traffic forecasting. IEEE Transactions on Intelligent Transportation Systems (2024)

2. Y. Huang, Z. Zhou, Z. Wang, X. Zhi, and X. Liu: Timesnet-pm2. 5: Interpretable timesnet for disentangling intraperiod and interperiod variations in pm2. 5 prediction. Atmosphere, 14(11), 1604 (2023)
3. J. Dong, H. Wu, H. Zhang, L. Zhang, J. Wang, and M. Long: Simmtm: A simple pre-training framework for masked time-series modeling. In: Advances in Neural Information Processing Systems, vol. 36 (2024).
4. J. A. Miller, M. Aldosari, F. Saeed, N. H. Barna, S. Rana, I. B. Arpinar, and N. Liu: A survey of deep learning and foundation models for time series forecasting. arXiv preprint arXiv:2401.13912 (2024).
5. H. Wu, J. Xu, J. Wang, and M. Long: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Advances in neural information processing systems, vol. 34, pp. 22419–22430 (2021).
6. Y. Zhang and J. Yan: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The eleventh international conference on learning representations (2023).
7. T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International conference on machine learning, pp. 27268–27286, PMLR (2022).
8. A. Zeng, M. Chen, L. Zhang, and Q. Xu: Are transformers effective for time series forecasting. In: Proceedings of the AAAI conference on artificial intelligence, vol. 37, pp. 11121–11128 (2023).
9. A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, and R. Yu: Long-term forecasting with tide: Time-series dense encoder. arXiv preprint arXiv:2304.08424 (2023).
10. H.-J. Y. Lu Han, Xu-Yang Chen and D.-C. Zhan: Softs: Efficient multivariate time series forecasting with series-core fusion. In: Advances in Neural Information Processing Systems (2024).
11. T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo: Reversible instance normalization for accurate time-series forecasting against distribution shift. In: International Conference on Learning Representations (2021).
12. W. Fan, P. Wang, D. Wang, D. Wang, Y. Zhou, and Y. Fu: Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In Proceedings of the AAAI conference on artificial intelligence, vol. 37, pp. 7522–7529 (2023).
13. Z. Liu, M. Cheng, Z. Li, Z. Huang, Q. Liu, Y. Xie, and E. Chen: Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. In: Advances in Neural Information Processing Systems, vol. 36 (2024).
14. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin: Attention is all you need. In Advances in neural information processing systems, vol. 30 (2017).
15. H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang: Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on artificial intelligence, vol. 35, pp. 11106–11115 (2021).
16. Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long: itransformer: Inverted transformers are effective for time series forecasting. In: The Twelfth International Conference on Learning Representations (2023).
17. K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu: Frequency-domain mlps are more effective learners in time series forecasting. In: Advances in Neural Information Processing Systems, vol. 36 (2024).
18. Z. Xu, A. Zeng, and Q. Xu: Fits: Modeling time series with 10k parameters. In The Twelfth International Conference on Learning Representations (2023).

19. H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long: Timesnet: Temporal 2d-variation modeling for general time series analysis. In: The Eleventh International Conference on Learning Representations (2022).
20. Y. Jia, Y. Lin, X. Hao, Y. Lin, S. Guo, and H. Wan: Witran: Water-wave information transmission and recurrent acceleration network for long-range time series forecasting. In: Advances in Neural Information Processing Systems, vol. 36 (2024).
21. Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, and C. Wang: Adarnn: Adaptive learning and forecasting of time series. In: Proceedings of the 30th ACM international conference on information & knowledge management, pp. 402–411 (2021).
22. Y. Liu, C. Li, J. Wang, and M. Long: Koopa: Learning non-stationary time series dynamics with koopman predictors. In: Advances in Neural Information Processing Systems, vol. 36 (2024).
23. Y. Liu, H. Wu, J. Wang, and M. Long: Non-stationary transformers: Exploring the stationarity in time series forecasting. In: Advances in Neural Information Processing Systems, vol. 35, pp. 9881–9893 (2022).
24. W. Fan, S. Zheng, P. Wang, R. Xie, J. Bian, and Y. Fu: Addressing distribution shift in time series forecasting with instance normalization flows. arXiv preprint arXiv:2401.16777 (2024).
25. L. Ardizzone, J. Kruse, C. Rother, and U. Köthe: Analyzing inverse problems with invertible neural networks. In: International Conference on Learning Representations (2018).
26. L. Dinh, J. Sohl-Dickstein, and S. Bengio: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016).
27. Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam: A time series is worth 64 words: Long-term forecasting with transformers. In: The Eleventh International Conference on Learning Representations (2008).
28. L. Van der Maaten and G. Hinton: Visualizing data using t-sne. Journal of machine learning research, vol. 9, no. 11 (2008).