A Notations

The notations is shown in table 1.

B Distribution shift in time series forecasting

Currently, most methods addressing distribution shift predominantly focus on images and static graphs, with limited research on distribution shift in time series forecasting [Zhou *et al.*, 2023]. In time series, data tends to exhibit temporal shift over time. For instance, the relationships between series may change over time, unseen data may emerge in the test set that was absent in the training set, and significant differences may exist between different lookback windows during the training process [Fan *et al.*, 2023]. Overall, distribution shift in time series are primarily attributed to the temporal evolution of data over time. This phenomenon can result in suboptimal performance of forecasting models, necessitating targeted approaches to handle the distribution shift that occur over time.

C More Experimental Details

C.1 More Dataset Details

We conduct our experiments on the following five real-world datasets:

- ***ECG***[1] This dataset is about Electrocardiogram(ECG) from the UCR time-series classification archive. It contains 140 nodes and each node has a length of 5000.
- ***Traffic***[2] This dataset contains hourly traffic data from 963 San Francisco freeway car lanes. The traffic data are collected since 2015/01/01 with the sampling interval of every 1 hour.
- ***WiKi***[3] This dataset contains a number of daily views of different Wikipedia articles and is collected from 2015/7/1 to 2016/12/31. It consists of approximately 145k time series and we randomly choose 2k from them as our experimental data set.
- ***Solar***[4] This dataset is about solar power collected by National Renewable Energy Laboratory. We choose the power plant data points in Florida as the data set which contains 593 points. The data is collected from 2006/01/01 to 2016/12/31 with the sampling interval of every 1 hour.
- ***COVID-19***[5] This dataset is about COVID-19 hospitalization in the U.S. states of California (CA) from 01/02/2020 to 31/12/2020 provided by the Johns Hopkins University with the sampling interval.

[1]http://www.timeseriesclassification.com/description.php?Dataset=ECG5000

[2]https://archive.ics.uci.edu/dataset/204/pems+sf

[3]https://www.kaggle.com/c/web-traffic-time-series-forecasting/data

[4]https://www.nrel.gov/grid/solar-power-data.html

[5]https://github.com/CSSEGISandData/COVID-19

C.2 Baselines

We conduct experiments on several representative and state-of-the-art baselines to verify the effectiveness of our DIAN.

- ***VAR***[6] VAR is a classic linear autoregressive model. We use the Statsmodels library which is a Python package that provides statistical computations to realize the VAR.
- ***DeepGLO***[7] DeepGLO models the relationships among variables by matrix factorization and employs a temporal convolution neural network to introduce non-linear relationships.We follow the recommended configuration as our experimental settings for wiki, electricity, and traffic datasets. For covid datasets, the vertical and horizontal batch size is set to 64, the rank of the global model is set to 64, the number 559 of channels is set to [32, 32, 32, 1], and the period is set to 7.
- ***LSTNet***[8] LSTNet use GCN and RNN to capture spatial and temporal dependencies for forecasting. We set the number of CNN hidden units is 100, the kernel size of the CNN layers is 4, the dropout is 0.2, the RNN hidden units is 100, the number of RNN hidden layers is 1, the learning rate is 0.001 and the optimizer is Adam.
- ***TCN***[9] TCN is a causal convolution model for regression prediction. We set the dropout is 0.25, the kernel size is 5, the number of hidden units is 150, the number of levels is 4 and the optimizer is Adam.
- ***Reformer***[10] Reformer is a Transformer-based model which can be executed efficiently on long sequences and with small memory use.
- ***Informer***[11] Informer uses self-attention mechanism to capture inter-variate dependencies. We set the dropout is 0.05, the number of encoder layers is 2, the number of decoder layers is 1, the learning rate is 0.0001, and the optimizer is Adam.
- ***Autoformer***[12] Autoformer proposes a decomposition architecture to progressively aggregate the long-term trend part from intermediate prediction. We set 2 encoder layers and 1 decoder layer.
- ***SFM***[13] SFM uses different frequency components based on LSTM model. We set the learning rate is 0.01, the frequency dimension is 10, the hidden dimension is 10 and the optimizer is RMSProp.
- ***StemGNN***[14] StemGNN leverages GFT and DFT to capture dependencies among variables in the frequency domain. We set the optimizer is RMSProp, the learning rate is 0.0001, the number of stacked layers is 5, and the dropout rate is 0.5.

[6]https://www.statsmodels.org/stable/index.html

[7]https://github.com/rajatsen91/deepglo

[8]https://github.com/laiguokun/LSTNet

[9]https://github.com/locuslab/TCN

[10]https://github.com/thuml/Autoformer

[11]https://github.com/zhouhaoyi/Informer2020

[12]https://github.com/thuml/Autoformer

[13]https://github.com/z331565360/State-Frequency-Memory-stock-prediction

[14]https://github.com/microsoft/StemGNN

Table 1: Notations.

Notations	Descriptions
X_t	the multivariate time series at timestamp t, $X \in R^{N\times 1}$
x_t^i	the value of multivariate time series at timestamp t, $x_t^i \in R^1$
A_t	adjacency matrix at timestamp t, $A_t \in R^{N\times N}$
$\mathcal{G}_t$	dynamic graph structure at timestamp t, $\mathcal{G}_t = (\mathbf{X}_{t-L+1:t}, A_t)$
$\mathbf{X}_{t-L+1:t}$	historical observations of length L at timestamp t, $\mathbf{X}_{t-L+1:t} \in R^{N\times L}$
$\mathbf{X}_{t+1:t+H}$	future value of length H at timestamp t, $\mathbf{X}_{t-L+1:t} \in R^{N\times H}$
$\widehat{\mathbf{X}}_{t+1:t+H}$	the predicted values of length H at timestamp t,$\mathbf{X}_{t-L+1:t} \in R^{N\times L}$
$\hat{\mathbf{V}}_{t+1:t+H}$	the predicted value using intervened samples
R_I^t, R_V^t	the spatial invariant and variant patterns
P_I^t, P_V^t	the temporal invariant and variant patterns
ϵ_1, ϵ_2	Errors in the prediction of functions g and y
$\mathcal{H}_{\mathcal{G}_t}, \mathcal{H}_{\mathbf{X}}^t$	the spatial and temporal overall representation, $\mathcal{H}_{\mathcal{G}_t} \in R^{1\times T\times d_c}$, $\mathcal{H}_{\mathbf{X}}^t \in R^{N\times 1\times d_c}$
$\mathrm{Attn}_I^{\mathcal{G}_t}(i), \mathrm{Attn}_V^{\mathcal{G}_t}(i)$	the spatial invariant and variant attention score for i-th series, $\mathrm{Attn}_I^{\mathcal{G}_t}(i), \mathrm{Attn}_V^{\mathcal{G}_t}(i) \in R^1$
$\mathbf{H}_I^{\ell+1}, \mathbf{H}_V^{\ell+1}$	the spatial invariant and variant embeddings of $(\ell+1)$-th layer, $\mathbf{H}_I^{\ell+1}, \mathbf{H}_V^{\ell+1} \in R^{N\times T\times d_c}$
$\mathbf{H}^{\ell+1}$	a new spatially-learned representation of the original time series
$\mathrm{Attn}_I^{\mathbf{H}_t^{\ell+1}}(t), \mathrm{Attn}_V^{\mathbf{H}_t^{\ell+1}}(t)$	the temporal invariant and variant attention score at timestamp t, $\mathrm{Attn}_I^{\mathbf{H}_t^{\ell+1}}(t), \mathrm{Attn}_V^{\mathbf{H}_t^{\ell+1}}(t) \in R^1$
$\mathbf{z}_I^t, \mathbf{z}_I^t$	the aggregations of temporal invariant and variant patterns
$\mathbf{h}^t$	the pattern summarization of invariant and variant aggregations
$s_I^{t_i}, s_V^{t_i}$	the intervened invariant and variant patterns
$\mathbf{h}_{inv}^{t_i}$	the intervened sample at t-th timestamp
$\mathrm{Aggr}_I, \mathrm{Aggr}_V$	the aggregation functions for temporal invariant and variant patterns
$\mathcal{F}_\Theta$	a function that maps historical observations to future predicted values
g	a function that learns from adjacency matrix, spatial invariant and variant patterns to get a new spatially-learned representation
y	a function that learns from temporal invariant and variant patterns to get a new temporally-learned representation

- ***MTGNN***[15] MTGNN aims to model inherent dependency relationships inter-seriesly. We set the parameter $load_static_feature$ to false. Regarding other parameters, we adopt the recommended settings.

- ***GraphWaveNet***[16] GraphWaveNet proopose the adaptive dependency matrix to learn spatial dependencies. We adopt the recommended configuration as our experimental settings where the learning rate is 0.001, the dropout is 0.3, the number of epochs is 50, and the optimizer is Adam.

- ***AGCRN***[17] AGCRN propose a data-adaptive graph to learn spatial dependencies and RNN to learn temporal dependencies. We set the embedding dimension is 10, the learning rate is 0.003, and the optimizer is Adam.

- ***TAMP-S2GCNets***[18] TAMP-S2GCNets explores the utility of MP to enhance knowledge representation mechanisms within the time-aware DL paradigm. We adopt the recommended configuration as our experimental settings.

- ***CoST***[19] CoST propose a contrastive learning framework to decouple season-trend representations. We set the representation dimension to 320, the learning rate to 0.001, and the batch size to 32.

[15]https://github.com/nnzhan/MTGNN

[16]https://github.com/nnzhan/Graph-WaveNet

[17]https://github.com/LeiBAI/AGCRN

[18]https://www.dropbox.com/sh/n0ajd5l0tdeyb80/AABGn-ejfV1YtRwjf_L0AOsNa?dl=0

[19]https://github.com/salesforce/CoST

C.3 Experimental Setting

We summarize the implementation details of our proposed DIAN as follows (The implementation details of baselines are introduced in C.2:

- ***Details of DIAN.***
Aggregation Functions. For invariant and variant patterns, we have different aggregation functions:

$$\mathbf{z}_I^t = Dropout(MLP(\mathbf{H}_t^{\ell+1} \times \mathrm{Attn}_I^{\mathbf{H}_t^{\ell+1}}(t) + \mathbf{H}_t^{\ell+1})) + \mathbf{H}_t^{\ell+1} \times \mathrm{Attn}_I^{\mathbf{H}_t^{\ell+1}}(t) + \mathbf{H}_t^{\ell+1}, \tag{1}$$

$$\mathbf{z}_V^t = Dropout(MLP(\mathbf{H}_t^{\ell+1} \times \mathrm{Attn}_V^{\mathbf{H}_t^{\ell+1}}(t))) + \mathbf{H}_t^{\ell+1} \times \mathrm{Attn}_V^{\mathbf{H}_t^{\ell+1}}(t), \tag{2}$$

SpatialProjection. The purpose of this function is to consolidate all series by mapping the dimension of series to 1 using a fully connected layer.

$$\mathrm{SpatialProjection} = Wx + b \tag{3}$$

TemporalProjection. Similar to SpatialProjection, the purpose of this function is to consolidate all timestamps by mapping the dimension of time to 1 using a fully connected layer.

$$\mathrm{TemporalProjection} = Wx + b \tag{4}$$

- ***Training details.*** We carefully tune the hyperparameters, including batch size, embedding size, spatial variant Engagement α, temporal variant Engagement β, intervened sample Engagementγ, and we choose the settings with the best performance for DIAN for different

datasets. Specifically, For Traffic and ECG dataset , we set the batch size as 64; for solar and Wiki dataset, the batch size is 32 and for COVID-19 dataset, the batch size is 16(limited by data scale). Except for WIKI dataset, the embedding size is set as 10 and for WIKI dataset, the embedding size is 16. For COVID-19 dataset, α is 0.55, β is 0.55, γ is 0; For ECG dataset, α is 0.3, β is 0.3, γ is 0.01; For Traffic dataset, α is 0.55, β is 0.55, γ is 0; For Solar dataset, α is 0.5, β is 0.3, γ is 0.01; For WIKI dataset, α is 0.5, β is 0.35, γ is 0.01.

- ***Evaluation Metrics*** We use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as evaluation matrics. Specifically, we have the ground truth at timestamp t, $y_t = \{X_{t+1}, \cdots, X_{t+H}\} \in R^{N \times H}$ and the predicted values $\hat{y}_t = \{\hat{X}_{t+1}, \cdots, \hat{X}_{t+H}\} \in R^{N \times H}$

$$MAE = \frac{1}{NH} \sum_{i=1}^{N} \sum_{j=1}^{H} |y_{ij} - \hat{y}_{ij}|, \tag{5}$$

$$RMSE = \sqrt{\frac{1}{NH} \sum_{i=1}^{N} \sum_{i=1}^{H} (y_{ij} - \hat{y_{ij}})^2)}, \tag{6}$$

where N is the series for evaluation, H is the length of horizon window.

D Additional Results

Table 2: Ablation study on Traffic dataset.

Metrics	Traffic		
	w/o spatial	w/o temporal	DIAN
MAE	0.014	0.014	**0.013**
RMSE	0.029	0.030	**0.029**

D.1 Ablation Study

To verify that each part of the model is valid, we also conducted ablation experiments on the Traffic dataset. The results are shown in Table2

As can be seen from the results, although the changes are relatively small, removing spatial or temporal modules will lead to increased errors.

E Further Analysis

Table 3: Performance comparison under different lookback window lengths on the COVID-19 dataset.

Lookback Window Length	COVID-19	
	MAE	RMSE
3	0.146	0.193
6	0.152	0.200
9	0.142	0.189
12	**0.128**	**0.175**

Table 4: Performance comparison under different embedding sizes on the COVID-19 dataset.

Embedding Size	COVID-19	
	MAE	RMSE
5	0.160	0.208
10	0.128	0.175
15	0.130	0.176
20	**0.131**	**0.175**

Table 5: Performance comparison under different intervened engagement on the ECG dataset.

Intervened Percentage	COVID-19	
	MAE	RMSE
0	0.050	0.076
0.01	0.049	0.075
0.1	0.053	0.080
0.5	0.050	0.076
1.0	0.050	0.076

E.1 Horizon Analysis.

The horizon window dictates the number of time points for prediction. A window that is too large might exceed the model's predictive capacity, leading to suboptimal performance, while a window that is too small might not fully showcase the model's capabilities. To verify DIAN has a stable predictive ability, with a fixed lookback of 12, we assessed DIAN's performance on the COVID-19 dataset as the horizon window varied from 3 to 12 and compare it with StemGNN, AGCRN, MTGNN, TAMP-S2GCNets, and CoST. The results are shown in table6, which demonstrates that DIAN outperforms other baselines and achieves an average 20.8%, and 19.1% improvement on MAE and RMSE respectively over the best baseline and indicate that a larger horizon window tends to result in increased error due to the introduction of more distribution changes, necessitating more complex modeling.

E.2 Lookback Analysis

The selection of the lookback window length determines the amount of information to be gathered, representing the length of the input sequence at a specific time stamp. A small lookback window leads to inadequate knowledge, resulting in poor model performance. Conversely, an excessively large lookback window increases the quantity of information, but it also introduces noise and error. Therefore, we conducted experiments using the COVID-19 dataset, fixing the horizon window length at 12, and assessed the appropriate setting of the lookback window by altering it from 3 to 12, ensuring sufficient prediction information was obtained from the COVID-19 dataset. The findings are presented in Table 3.

The results indicate that the optimal performance occurs with a lookback window of 12 and the poorest performance is observed with a lookback window of 6. This result indicates that for the COVID-19 dataset, when the lookback window length is set to 12, the gain in extracting useful information outweighs the increase in noise interference. However, for lookback window lengths of 3, 6, and 9, the limited amount of

Table 6: Performance comparison under different horizon window lengths on the COVID-19 dataset.

Models \ Horizon Window Length	3		6		9		12	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
StemGNN	0.247	0.318	0.344	0.429	0.359	0.442	0.421	0.508
AGCRN	0.130	<u>0.172</u>	0.171	0.218	0.224	0.277	0.254	0.309
MTGNN	0.276	0.379	0.446	0.513	0.484	0.548	0.394	0.488
TAMP-S2GCNets	0.140	0.190	<u>0.150</u>	<u>0.200</u>	<u>0.170</u>	<u>0.230</u>	<u>0.180</u>	<u>0.230</u>
CoST	<u>0.122</u>	0.246	0.157	0.318	0.183	0.364	0.202	0.377
DIAN(ours)	**0.110**	**0.152**	**0.116**	**0.162**	**0.133**	**0.180**	**0.128**	**0.175**

Table 7: Pattern study on Solar and COVID-19 dataset.

Metrics	Solar			COVID-19		
	w/o S-Variant	w/o T-Variant	DIAN	w/o S-Variant	w/o T-Variant	DIAN
MAE	0.101	0.099	**0.095**	0.151	0.144	**0.128**
RMSE	0.179	0.178	**0.167**	0.199	0.192	**0.175**

historical data prevents the discovery of underlying patterns, resulting in larger errors.

E.3 Embedding Size Analysis

We conduct an experiment about the embedding size on COVID-19 dataset. If the embedding size is too small, the feature learning may be incomplete, resulting in poor performance of the model. However, if it is too large, it will lead to overfitting and increase the error. We set the embedding size from 5 to 20 to observe MAE and RMSE of the results. The results are shown in Table4.

It can be seen from the results that the best performance is when the embedding size is equal to 10, which indicates that for a small data set such as COVID-19, the embedding size set of 10 is enough to learn the knowledge required for prediction. If the embedding size continues to expand, overfitting will occur.

E.4 Pattern Study.

Some previous researchers have been accustomed to relying solely on invariant patterns for prediction, disregarding variant patterns. However, this approach can lead to information loss and subsequently result in a decrease in model accuracy. In our study, we conducted experiments on the COVID-19 and Solar datasets to validate the contribution of variant patterns to predictions. Specifically, we compared the performance of models that excluded variant patterns in Temporal Invariant-Variant Attention or Decoupled Invariant-Variant Convolution with the model that utilized both variant and invariant patterns for forecasting. The results are shown in table7. From the results, it is evident that removing variant patterns, either in the Temporal Invariant-Variant Attention or Decoupled Invariant-Variant Convolution modules, leads to a decrease in accuracy. This indicates that variant patterns contain valuable information, and therefore, we should consider them in our predictions.

E.5 Intervention Analysis

We use data-augmentation methods to counter potential distribution shift. Specifically, we construct new intervened samples using computed temporal variant patterns as recombinations. If too many intervened samples are introduced, the learning of the original sequence will be biased. If too little is introduced, the future distribution shift cannot be handled and the model becomes less adaptable. Therefore, we tested on the COVID-19 dataset by setting the intervened ratio from 0.01 to 1. The result are shown in Table5. It can be seen from the results that the model performs best when the intervened sample is 0.01, indicating that the greatest gains can be made when considering the potential distribution shift of 0.01.

Acknowledgments

This research is funded by the Science and Technology Development Fund (FDCT), Macau SAR (file no. 0123/2023/RIA2, 001/2024/SKL), the Start-up Research Grant of University of Macau (File no. SRG2021-00017-IOTSC).

References

[Fan *et al.*, 2023] Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-ts: A general paradigm for alleviating distribution shift in time series forecasting, 2023.

[Zhou *et al.*, 2023] Zhengyang Zhou, Qihe Huang, Kuo Yang, Kun Wang, Xu Wang, Yudong Zhang, Yuxuan Liang, Yang Wang, Kunwang, XuWang, Yudong, and Zhang. Maintaining the status quo: Capturing invariant relations for ood spatiotemporal learning. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.