Adaptive Normalization for Non-stationary Time Series Forecasting: A Temporal Slice Perspective

Zhiding Liu[1,2], Mingyue Cheng[1,2], Zhi Li[3], Zhenya Huang[1,2], Qi Liu[1,2], Yanhu Xie[4], Enhong Chen[1,2*]

[1]Anhui Province Key Laboratory of Big Data Analysis and Application,
University of Science and Technology of China
[2]State Key Laboratory of Cognitive Intelligence
[3]Shenzhen International Graduate School
[4]The First Affiliated Hospital of University of Science and Technology of China

Presenter: Zhiding Liu

Introduction

- What is time series?
  - ***Any signals*** collected in chronological order.
  - Ubiquitous/Noisy & Chaotic/Extremely long/Multivariate
- What is forecasting?
  - Given the observation of past S time steps, predict the values of future T steps.
  - Temporal/Channel dependence
- Why forecasting?
  - Weather report
  - Healthcare analysis
  - Decision making
  - …

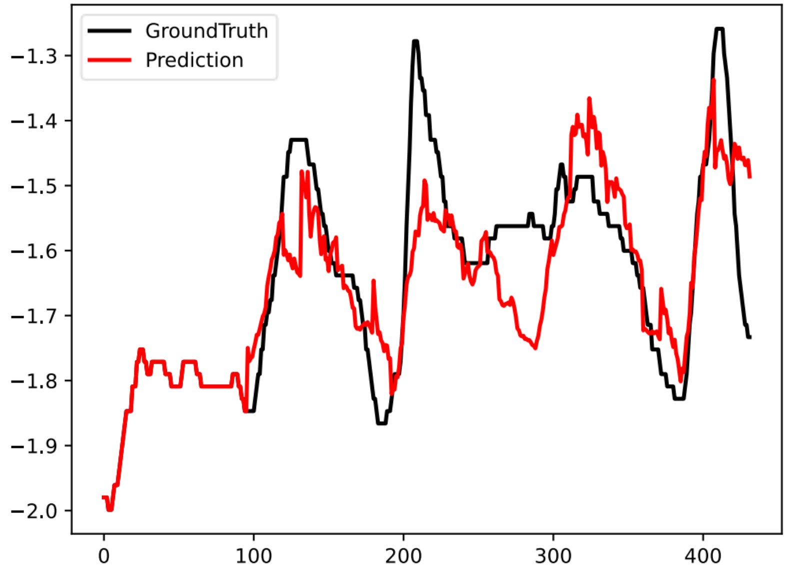


Figure 1: Illustration of forecasting.

Introduction

- **Tremendous efforts** have been devoted in **designing powerful networks** and therefore greatly advanced the accuracy of forecasting performance.

- The intrinsic **non-stationary property** hinders the generalizability of deep-learning-based models.
  - The distribution shift in time series, i.e., $\forall_{i,j}\ p(x^i) \neq p(x^j)$
  - Existing forecasting methods barely rely on the non-linear capacity to tackle the challenge.

- To **explicitly alleviate the impact of non-stationarity**, adaptive normalization on input series is the feasible and mainstream solution.
  - Ensure the inputs are ***I.I.D.*** through normalization.

Methodology

- Motivation
  - Existing normalization methods are based on an assumption that ***the input series of an instance follows the same distribution***.
    - In the real-world scenarios, time series points rapidly change over time. For any given time slices of instance k, x_i^k, x_j^k, $p(x_i^k) \neq p(x_j^k)$.
  - Previous works either ***ignore*** the restoration of non-stationary information or simply ***adopt the statistical properties of input series to denormalize the output results***.
    - Lead to a prediction shift of the final forecasting results due to a bad estimation of future statistics.
- Our thoughts
  - Split series into non-overlap equally-sized slices and model the **local-region non-stationarity** under them.
  - Employ a **statistics prediction module** learning to estimate the distribution of future slices precisely.

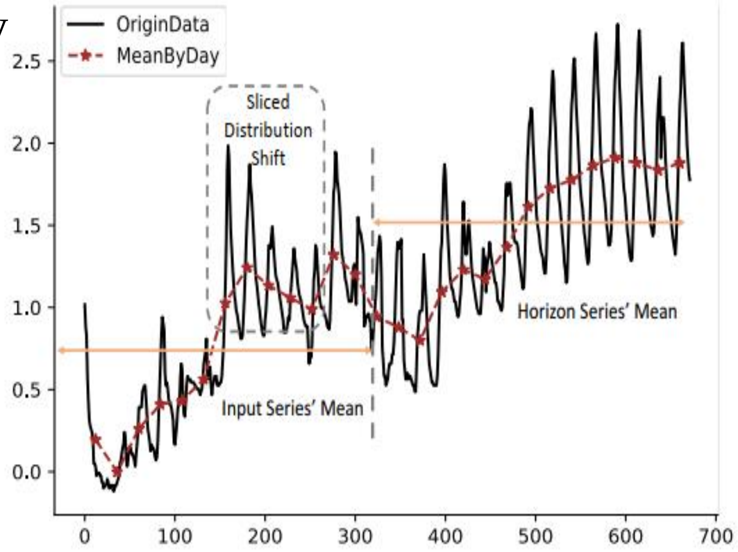


Figure 2: A forecasting instance with non-stationarity.

Methodology

- Sliced normalization
  - Removing the local non-stationarity for each time slice according to their statistics.

$$\mu_j^i = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_{j,t}^i,\ (\sigma_j^i)^2 = \frac{1}{T}\sum_{t=1}^{T}(\boldsymbol{x}_{j,t}^i - \mu_j^i)^2, \qquad \bar{\boldsymbol{x}}_j^i = \frac{1}{\sigma_j^i + \epsilon}\cdot(\boldsymbol{x}_j^i - \mu_j^i).$$

- Statistics prediction
  - Under our assumption, a natural challenge is that how to **estimate the evolving distributions** for each future slice.
  - We adopt a two-layer perceptron network responsible for this task for simplicity and efficiency.

$$\hat{\boldsymbol{\mu}}^i = \boxed{\mathbf{W}_1} * MLP(\boxed{\boldsymbol{\mu}^i - \rho^i, \bar{\boldsymbol{x}}^i - \rho^i}) + \boxed{\mathbf{W}_2} * \rho^i,$$
$$\hat{\boldsymbol{\sigma}}^i = MLP(\boldsymbol{\sigma}^i, \bar{\boldsymbol{x}}^i).$$

$\boxed{}$: residual learning

$\boxed{}$: individual preference

- The overall mean of the input sequence is a *maximum likelihood estimation* of the target sequence's mean → **residual learning**.
- Different variables may exhibit distinct patterns in scale changes → **individual preference**.

Methodology

- Sliced de-normalization
  - The non-stationarity information is vital for forecasting.
  - Restore them into predicted results in a slice perspective.

$$\hat{\boldsymbol{y}}_j^i = \bar{\boldsymbol{y}}_j^i * (\hat{\boldsymbol{\sigma}}_j^i + \epsilon) + \hat{\boldsymbol{\mu}}_j^i.$$

- **S**licing **A**daptive **N**ormalization (SAN)

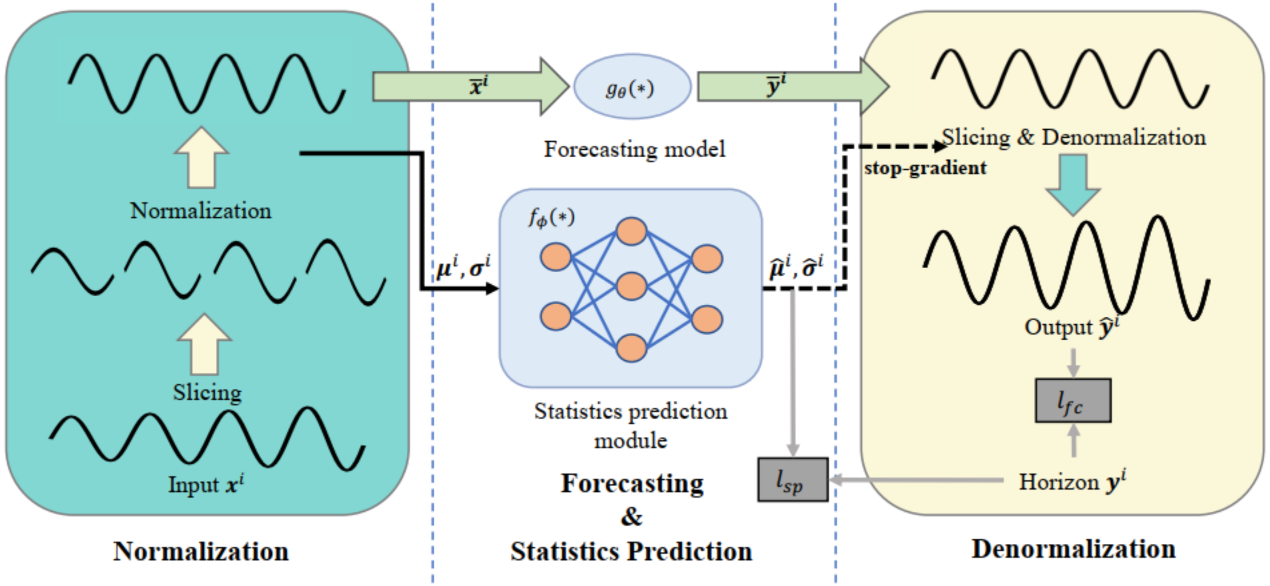


Figure 3: Illustration of the proposed SAN framework.

Methodology

- Two-stage training schema
  - It forms to a bi-level optimization problem when joint training SAN and backbone model, as the target is to ensure the similarity between distributions of **denormalized output and ground truth**.

$$arg\ min_{\theta} \sum_{(\boldsymbol{x}^i, \boldsymbol{y}^i)} l_{fc}(\theta, \phi^*, (\boldsymbol{x}^i, \boldsymbol{y}^i)),$$
$$s.t. \phi^* = arg\ min_{\phi} \sum_{(\boldsymbol{x}^i, \boldsymbol{y}^i)} l_{sp}(\theta, \phi, (\boldsymbol{x}^i, \boldsymbol{y}^i)).$$

  - Relax the optimization objective of statistic prediction module to **estimating the future distribution.**
    - The original non-stationary forecasting task is divided into **decoupled** statistic prediction task and stationary forecasting task.
  - Qualities:
    - **Simplifies** the task of non-stationary forecasting through **divide-and-conquer**.
    - Estimate **more accurately** on future distributions.

Experiments

NEURAL INFORMATION PROCESSING SYSTEMS

- Setup
  - The widely used benchmark with 9 datasets:

Dataset	Variables	Sampling Frequency	Length	Slicing Length	ADF*
Electricity	321	1 Hour	26,304	24	-8.44
Exchange	8	1 Day	7,588	6	-1.90
Traffic	862	1 Hour	17,544	24	-15.02
Weather	21	10 Minutes	52,696	12	-26.68
ILI	7	1 Week	966	6	-5.33
ETTh1&ETTh2	7	1 Hour	17,420	24	-5.91&-4.13
ETTm1&ETTm2	7	15 Minutes	69,680	12	-14.98&-5.66

* A smaller ADF test result indicates a more stationary time series data

  - Baseline models:
    - RevIN (ICLR’22), NST(NIPS’22), Dish-TS(AAAI’23)
  - Backbone models:
    - Autoformer(NIPS’21),FEDformer(ICML’22),SCINet(NIPS’22),DLinear(AAAI’23)
    - Slice-based models: PatchTST(ICLR’23),Crossformer(ICLR’23)

Results

- Overall forecasting performance on SAN-enhanced backbone models.

Dataset	Horizon	Dlinear MSE	Dlinear MAE	+ SAN MSE	+ SAN MAE	FEDformer MSE	FEDformer MAE	+ SAN MSE	+ SAN MAE	Autoformer MSE	Autoformer MAE	+ SAN MSE	+ SAN MAE	SCINet MSE	SCINet MAE	+ SAN MSE	+ SAN MAE
Electricity	96	0.140	0.237	**0.137**	**0.234**	0.185	0.300	**0.164**	**0.272**	0.195	0.309	**0.172**	**0.281**	0.213	0.316	**0.152**	**0.256**
	192	0.153	0.250	**0.151**	**0.247**	0.196	0.310	**0.179**	**0.286**	0.215	0.325	**0.195**	**0.300**	0.224	0.329	**0.163**	**0.266**
	336	0.168	0.267	**0.166**	**0.264**	0.215	0.330	**0.191**	**0.299**	0.237	0.344	**0.211**	**0.316**	0.230	0.334	**0.178**	**0.283**
	720	0.203	0.301	**0.201**	**0.295**	0.244	0.352	**0.230**	**0.334**	0.292	0.375	**0.236**	**0.335**	0.260	0.356	**0.206**	**0.307**
Exchange	96	0.086	**0.213**	**0.085**	0.214	0.152	0.281	**0.079**	**0.205**	0.152	0.283	**0.082**	**0.208**	0.126	0.269	**0.082**	**0.200**
	192	**0.161**	**0.297**	0.177	0.317	0.273	0.380	**0.156**	**0.295**	0.369	0.437	**0.157**	**0.296**	0.266	0.392	**0.169**	**0.293**
	336	0.338	0.437	**0.294**	**0.407**	0.452	0.498	**0.260**	**0.384**	0.534	0.544	**0.262**	**0.385**	0.574	0.541	**0.320**	**0.409**
	720	0.999	0.755	**0.726**	**0.649**	1.151	0.830	**0.697**	**0.633**	1.222	0.848	**0.689**	**0.629**	1.136	0.818	**0.892**	**0.712**
Traffic	96	**0.411**	**0.283**	0.412	0.288	0.579	0.363	**0.536**	**0.330**	0.654	0.403	**0.569**	**0.350**	0.626	0.393	**0.542**	**0.344**
	192	**0.423**	**0.289**	0.429	0.297	0.608	0.376	**0.565**	**0.345**	0.654	0.410	**0.594**	**0.364**	0.613	0.396	**0.545**	**0.358**
	336	**0.437**	**0.297**	0.445	0.306	0.620	0.385	**0.580**	**0.354**	0.629	0.391	**0.591**	**0.363**	0.625	0.398	**0.563**	**0.369**
	720	**0.467**	**0.316**	0.474	0.319	0.630	0.387	**0.607**	**0.367**	0.657	0.402	**0.623**	**0.380**	0.639	0.409	**0.607**	**0.381**
Weather	96	0.175	0.237	**0.152**	**0.210**	0.246	0.328	**0.179**	**0.239**	0.247	0.320	**0.194**	**0.256**	0.181	0.260	**0.169**	**0.232**
	192	0.217	0.275	**0.196**	**0.254**	0.281	0.341	**0.234**	**0.296**	0.302	0.361	**0.258**	**0.316**	0.239	0.311	**0.215**	**0.275**
	336	0.263	0.314	**0.246**	**0.294**	0.337	0.376	**0.304**	**0.348**	0.362	0.394	**0.329**	**0.367**	0.293	0.348	**0.267**	**0.314**
	720	0.325	0.366	**0.315**	**0.346**	0.414	0.426	**0.400**	**0.404**	**0.427**	**0.433**	0.440	0.438	0.345	0.380	**0.338**	**0.365**
ILI	24	2.297	1.055	**2.122**	**1.001**	3.205	1.255	**2.614**	**1.119**	3.309	1.270	**2.777**	**1.157**	7.467	2.039	**2.776**	**1.163**
	36	2.323	1.070	**2.029**	**0.978**	3.148	1.288	**2.537**	**1.079**	3.207	1.216	**2.649**	**1.104**	7.035	1.948	**2.411**	**1.026**
	48	2.262	1.065	**2.041**	**0.971**	2.913	1.168	**2.416**	**1.032**	3.166	1.198	**2.420**	**1.029**	7.225	1.955	**2.295**	**1.004**
	60	2.443	1.124	**2.089**	**0.973**	2.853	1.161	**2.299**	**1.003**	2.947	1.159	**2.401**	**1.021**	7.335	1.957	**2.487**	**1.063**
ETTh2	96	0.292	0.356	**0.277**	**0.338**	0.341	0.382	**0.300**	**0.355**	0.384	0.420	**0.316**	**0.366**	0.690	0.625	**0.294**	**0.347**
	192	0.383	0.418	**0.340**	**0.378**	0.426	0.436	**0.392**	**0.413**	0.457	0.454	**0.413**	**0.426**	0.991	0.742	**0.374**	**0.398**
	336	0.473	0.477	**0.356**	**0.398**	0.481	0.479	**0.459**	**0.462**	0.468	0.473	**0.446**	**0.457**	1.028	0.759	**0.412**	**0.430**
	720	0.708	0.599	**0.396**	**0.435**	**0.458**	0.477	0.462	**0.472**	0.473	0.485	**0.471**	**0.474**	1.363	0.885	**0.437**	**0.461**

Dataset	Horizon	PatchTST MSE	PatchTST MAE	+ SAN MSE	+ SAN MAE	Crossformer MSE	Crossformer MAE	+ SAN MSE	+ SAN MAE
Electricity	96	0.138	**0.233**	**0.136**	0.234	0.150	0.258	**0.143**	**0.246**
	192	0.153	**0.247**	**0.150**	**0.247**	0.175	0.284	**0.162**	**0.265**
	336	0.170	**0.263**	**0.165**	0.264	0.218	0.325	**0.177**	**0.280**
	720	0.206	**0.296**	**0.200**	**0.296**	0.226	0.324	**0.221**	**0.318**
Exchange	96	0.094	**0.216**	**0.087**	0.218	0.283	0.393	**0.087**	**0.219**
	192	0.191	0.311	**0.181**	**0.323**	1.087	0.804	**0.171**	**0.313**
	336	0.343	0.427	**0.305**	**0.418**	1.367	0.905	**0.286**	**0.401**
	720	0.888	0.706	**0.659**	**0.620**	1.546	0.987	**0.749**	**0.653**
Weather	96	**0.147**	**0.197**	0.150	0.205	**0.148**	0.214	0.151	**0.210**
	192	**0.191**	**0.240**	0.194	0.252	0.201	0.270	**0.198**	**0.253**
	336	0.244	**0.282**	**0.243**	0.290	**0.248**	0.311	**0.248**	**0.294**
	720	0.320	**0.334**	**0.311**	0.343	0.366	0.395	**0.322**	**0.350**
ETTh1	96	0.382	0.403	**0.375**	**0.398**	0.390	0.417	**0.387**	**0.402**
	192	0.416	0.423	**0.413**	**0.422**	0.424	0.448	**0.413**	**0.425**
	336	0.441	0.440	**0.428**	**0.434**	0.486	0.492	**0.436**	**0.431**
	720	0.470	0.475	**0.445**	**0.461**	0.507	0.519	**0.467**	**0.474**
ETTm2	96	0.174	0.261	**0.167**	**0.260**	0.330	0.401	**0.170**	**0.262**
	192	0.238	0.307	**0.222**	**0.298**	0.623	0.543	**0.224**	**0.301**
	336	0.293	0.346	**0.276**	**0.334**	0.887	0.637	**0.274**	**0.333**
	720	0.373	0.401	**0.366**	**0.393**	0.844	0.640	**0.366**	**0.390**

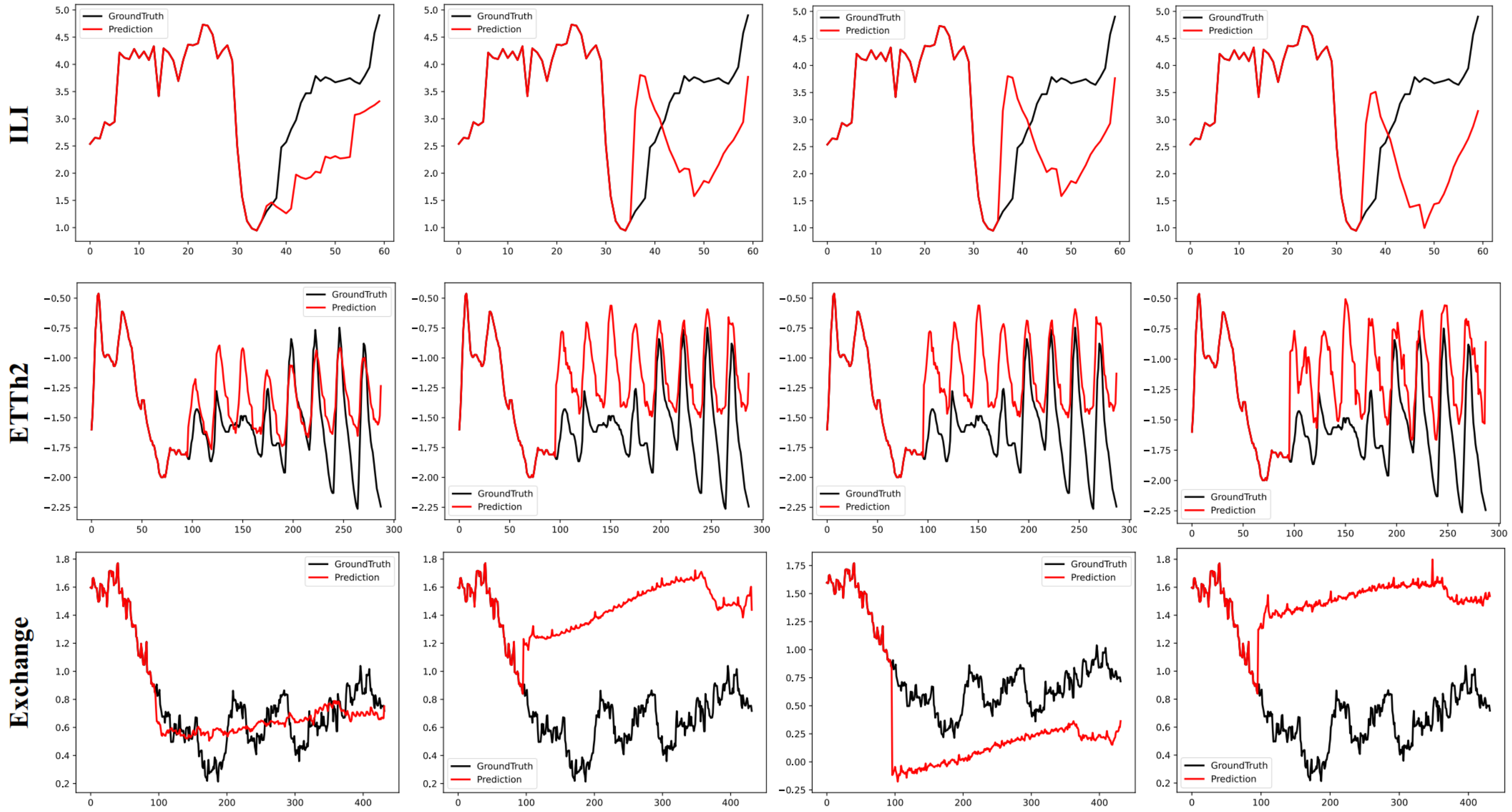
Experiments

- Results
  - Comparison between SAN and existing normalization approaches.

Methods	FEDformer +SAN	FEDformer +RevIN	FEDformer +NST	FEDformer +Dish-TS	FEDformer IMP(%)	Autoformer +SAN	Autoformer +RevIN	Autoformer +NST	Autoformer +Dish-TS	Autoformer IMP(%)
Electricity	**0.191**	0.200	0.198	0.203	3.54	**0.204**	0.219	0.213	0.231	4.23
Exchange	**0.298**	0.474	0.480	0.704	37.13	**0.297**	0.495	0.494	1.008	39.88
Traffic	**0.572**	0.647	0.649	0.652	11.59	**0.594**	0.666	0.664	0.677	10.54
Weather	0.279	0.268	**0.267**	0.398	-4.49	0.305	**0.290**	**0.290**	0.433	-5.17
ILI	**2.467**	2.962	3.084	2.846	13.32	**2.562**	3.151	3.235	3.180	18.69
ETTh1	**0.447**	0.463	0.456	0.461	1.97	**0.518**	0.519	0.521	0.521	0.19
ETTh2	**0.404**	0.465	0.481	1.004	13.12	**0.411**	0.489	0.465	1.175	11.61
ETTm1	**0.377**	0.415	0.411	0.422	8.27	**0.406**	0.562	0.535	0.567	24.11
ETTm2	**0.287**	0.310	0.315	0.759	7.42	**0.311**	0.325	0.331	0.894	4.31

Experiments

- Results



(a) SAN (b) RevIN (c) NST (d) Dish-TS

Figure 4: Visualization of forecasting results comparing SAN and baseline models.

Conclusion

NEURAL INFORMATION PROCESSING SYSTEMS

- We focused on alleviating the **non-stationary property** of time series data using a novel **slice view** in the forecasting task.
- We proposed **S**licing **A**daptive **N**ormalization (SAN)
  - A **model-agnostic** approach that removes the non-stationary factors the input by normalization and restores them to the output through denormalization **on a per-slice basis**.
  - With a two-stage training schema for the statistics prediction module, SAN **simplifies the non-stationary forecasting task through divide and conquer**.
  - Compared to existing normalization methods, SAN could better alleviate **the local-region non-stationarity** and provides more **accurate estimation on future distributions**.
- Extensive experiments validated the effectiveness of our method.

Thanks

https://github.com/icantnamemyself/SAN

zhiding@mail.ustc.edu.cn