Channel-Aware Low-Rank Adaptation in Time Series Forecasting

Tong Nie[†]
Tongji University
Shanghai, China
nietong@tongji.edu.cn

Yuewen Mei
Tongji University
Shanghai, China
meiyuewen@tongji.edu.cn

Guoyang Qin
Tongji University
Shanghai, China
2015qgy@tongji.edu.cn

Jian Sun
Tongji University
Shanghai, China
sunjian@tongji.edu.cn

Wei Ma[‡]
The Hong Kong Polytechnic University
Hong Kong SAR, China
wei.w.ma@polyu.edu.hk

ABSTRACT

The balance between model capacity and generalization has been a key focus of recent discussions in long-term time series forecasting. Two representative channel strategies are closely associated with model expressivity and robustness, including channel independence (CI) and channel dependence (CD). The former adopts individual channel treatment and has been shown to be more robust to distribution shifts, but lacks sufficient capacity to model meaningful channel interactions. The latter is more expressive for representing complex cross-channel dependencies, but is prone to overfitting. To balance the two strategies, we present a channel-aware low-rank adaptation method to condition CD models on identity-aware individual components. As a plug-in solution, it is adaptable for a wide range of backbone architectures. Extensive experiments show that it can consistently and significantly improve the performance of both CI and CD models with demonstrated efficiency and flexibility. The code is available at https://github.com/tongnie/C-LoRA.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

Long-Term Time Series Forecasting, Low-Rank Adaptation, Channel Independence, Channel Mixing

ACM Reference Format:

Tong Nie, Yuewen Mei, Guoyang Qin, Jian Sun, and Wei Ma. 2024. Channel-Aware Low-Rank Adaptation in Time Series Forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3679884

1 INTRODUCTION

As a fundamental scientific problem in diverse fields such as weather, economics, energy, and transportation, multivariate time series forecasting has attracted great interest in recent years. Especially, advanced neural forecasting architectures are designed to address the challenging long-term series forecasting (LTSF) problem, and two branches of methods, including Transformer- and MLP-based models, have achieved remarkable progress [2, 9, 11, 15, 17, 19–21].

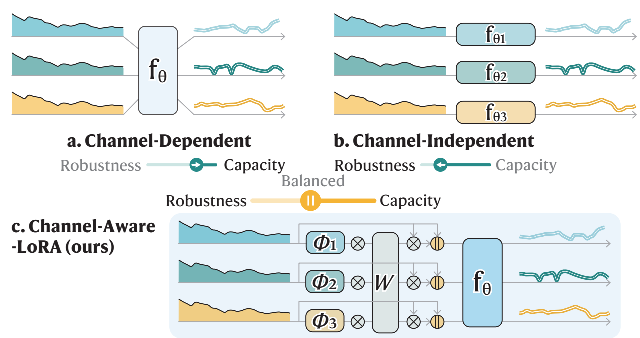

Figure 1: The proposed channel-aware low-rank adaptation.

By exploring the two types of architectures, recent studies have identified the core challenges faced in LTSF: *the balance between generalization and model capacity* [4]. On the one hand, real-world time series can be non-stationary, manifesting as a distribution shift in both training and test data. For example, the hourly energy consumption can vary in different workdays and seasons. Models need to be robust to confront the biased distribution. On the other hand, LTSF models are required to have enough capacity to represent the interweaved relationships between channels and possibly heterogeneous channel patterns. This frequently occurs in real-world time series, such as the weather and traffic system.

To address these challenges, very recent studies have shifted the focus from architecture designs to discussions of effective channel strategies [1, 9, 13, 20], as channel management methods can impact both robustness and expressivity. Generally, there are two primary channel methods: Channel-Independent (CI) and Channel-Dependent (CD) strategies. CD models treat series in all channels as a whole and employ a global predictor to model channel relations implicitly or explicitly. In contrast, CI strategy views multivariate channels as multiple univariate series to consider fine-grained individual patterns and overlooks potential channel interactions. Empirical results have shown that CI models are more robust in addressing distribution shift but have too expensive hypothesis spaces; while CD models have larger capacities but are prone to overfit and less flexible in modeling channel-specific phenomena.

[†] Also with The Hong Kong Polytechnic University. [‡] Corresponding author.

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10.
https://doi.org/10.1145/3627673.3679884

Limitation of existing approaches. The above dilemma calls for a new channel strategy that can balance CI and CD methods. Several pioneering approaches are leading the way in this field, such as group-aware embedding [18] and inverted embedding [9] for Transformers, leading indicator estimation [22], and channel clustering [1]. However, they are either limited to specific types of backbone model, or hard to improve the performance by a large margin without significantly increasing the computational burden.

Our solution. Inspired by recent progress in low-rank adaptation [5], we propose a channel-aware low-rank adaptation (C-LoRA) method to achieve a trade-off between the two strategies and provide an alternative in a parameter-efficient way. Specifically, we parameterize each channel a low-rank factorized adapter to consider individual treatment. Then the specialized channel adaptation is conditioned on the series information to form an identity-aware embedding. The cross-channel relational dependencies are simultaneously exploited by combining a globally shared CD model.

Contribution. C-LoRA is a plug-in solution that is seamlessly adaptable to a wide range of SOTA time series models. It contains almost no changes to the existing architecture. Extensive experiments demonstrate that it can consistently improve the performance of both CD and CI backbones by a large margin. Moreover, it has great efficiency, flexibility to transfer across datasets, and can enhance channel identity for accurate channel interaction modeling.

2 PRELIMINARIES

Multivariate Time Series Forecasting. Given the historical set $\mathbf{X} = \{\mathbf{x}_1, \dots \mathbf{x}_T\} \in \mathbb{R}^{T\times C}$, where T is the look-back window and C is the number of channels (variates), our objective is to learn a predictive function f_θ to estimate the future set $\mathbf{Y} = \{\mathbf{x}_{T+1}, \dots, \mathbf{x}_{T+H}\} \in \mathbb{R}^{H\times C}$ with a horizon H. For brevity, we denote $\mathbf{X}_{:,c} \in \mathbb{R}^T$ as the c-th channel and $\mathbf{X}_{t,:} \in \mathbb{R}^C$ as the observations at time t.

Channel-Dependent (CD). CD strategy is widely adopted in multivariate time series forecasting. CD models treat series in all channels as a whole and employ a global predictor $f_\theta : \mathbb{R}^{T\times C} \mapsto \mathbb{R}^{H\times C}$ to simultaneously forecast all components. In particular, a special case is called *channel mixing (CM)*, which models the explicit channel information interaction either by MLPs or self-attention. Given the training set $\{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^N$, the global CD loss is given by:

$$\theta^* = \arg\min_\theta \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{Y}^{(i)} - f_\theta(\mathbf{X}^{(i)})\|_F^2, \tag{1}$$

where N is the number of training samples.

Channel-Independent (CI). In contrast, CI strategy views multivariate channels as multiple univariate time series and ignores channel interactions. Particularly, the *channel-individual (CInd)* model is a special case that assigns a unique local predictor for each channel: $f_{\theta_i} : \mathbb{R}^T \mapsto \mathbb{R}^H, \forall i = 1, \dots, C$. The CI loss is given by:

$$\theta_1^*, \dots, \theta_C^* = \arg\min_{\theta_1,\dots,\theta_C} \frac{1}{NC}\sum_{i=1}^{N}\sum_{c=1}^{C} \|\mathbf{Y}_{:,c}^{(i)} - f_{\theta_c}(\mathbf{X}_{:,c}^{(i)})\|_F^2. \tag{2}$$

3 METHODOLOGY

As stated above, CI models are more robust in addressing distribution shift but have too expensive hypothesis spaces; while CD models have larger model capacities but are prone to overfit [4]. This section proposes the channel-aware low-rank adaptation model to balance channel-specific treatment and channel-wise dependence.

3.1 General Forecasting Backbone

As an efficient plugin, C-LoRA can be easily incorporated into a wide range of forecasting architectures. We first introduce a general forecasting template for both the CI and CD models as follows.

$$\begin{aligned}
\overline{\mathbf{X}} &= \textsc{Normalization}(\mathbf{X}),\\
\mathbf{z}_c^{(0)} &= \textsc{TokenEmbedding}(\overline{\mathbf{X}}_{:,c}), \forall c = 1, \dots, C,\\
\text{(Optional)}: \mathbf{Z}^{(\ell+1)} &= \textsc{ChannelMixing}(\mathbf{Z}^{(\ell)}), \forall \ell = 0, \dots, L,\\
\widehat{\mathbf{Y}} &= \textsc{Projection}(\mathbf{Z}^{(L+1)}),
\end{aligned} \tag{3}$$

where Normalization such as RevIN [6] is adopted to address the nonstationarity of time series, TokenEmbedding : $\mathbb{R}^T \mapsto \mathbb{R}^D$ and Projection : $\mathbb{R}^D \mapsto \mathbb{R}^H$ are usually implemented by MLPs to process temporal features, and ChannelMixing : $\mathbb{R}^{C\times D} \mapsto \mathbb{R}^{C\times D}$ is optional for CD models by Transformer blocks or MLPs.

Many state-of-the-art forecasting frameworks follow this template, such as TSMixer [2], RMLP [7], iTransformer [9], and FreTS [19]. For other Transformer-based architectures that employ token-wise attention, such as Autoformer [17] and Informer [23], we can adapt them with a simple inverting strategy [9] and adopt C-LoRA.

3.2 Channel-Aware Low-Rank Adaptation

We elaborate C-LoRA by first revisiting the two strategies. To account for channel-specific effects, CI (or CInd) methods can adopt individual models for each channel, which instantiate the TokenEmbedding with a series of mappings, e.g., different MLPs:

$$\mathbf{z}_c^{(0)} = \mathsf{MLP}_c(\overline{\mathbf{X}}_{:,c}; \theta_c), \forall c = 1, \dots, C, \tag{4}$$

then the hypothesis class of all individuals is $\mathcal{H}_{\text{CI}} = \{\mathsf{MLP}_c(\cdot; \theta_c) | \theta_c \in \Theta, c = 1, \dots, C\}$ where Θ is the parameter space. However, such a hypothesis class is computationally expensive, and pure CI models fail to exploit multivariate correlational structures [3].

In contrast, the CD strategy is more expressive by modeling channel interactions either explicitly with ChannelMixing or implicitly by optimizing the global loss in Eq. (1). However, they can have difficulty capturing individual channel patterns with a shared encoder $\mathsf{MLP}(\overline{\mathbf{X}}; \theta)$, and the CM operation can generate mixed channel identity information, causing an indistinguishment issue [11].

Combining the merits of the two strategies, we propose a novel strategy that considers channel-wise adaptation in a CD model. Specifically, to model individual channels in a parameter-efficient way, a low-rank *adapter* is specialized for each channel $\phi^{(c)} \in \mathbb{R}^{r\times D}$, where $r \ll D$ is the intrinsic rank. Then we can condition on another low-rank matrix to project it to a larger dimension:

$$\widetilde{\phi}^{(c)} = \mathsf{ReLU}(\phi^{(c),\mathsf{T}}\mathbf{W}) \in \mathbb{R}^{D\times d}, \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{r\times d}$ and d is the adaptation dimension. $\widetilde{\phi}^{(c)}$ characterizes channel-specific parameters, and it needs to be aware of the series information to consider the channel identity. Then we have:

$$\mathbf{z}_{c,\phi}^{(0)} = \mathbf{z}_c^{(0),\mathsf{T}}\widetilde{\phi}^{(c)} \in \mathbb{R}^d, \tag{6}$$

where $\mathbf{z}_c^{(0)} = \mathsf{MLP}(\overline{\mathbf{X}}_{:,c}; \theta)$ is obtained by a CD model shared by all channels. By aggregating all channel adaptations $\mathbf{Z}_\phi^{(0)} = \{\mathbf{z}_{c,\phi}^{(0)}\}_{c=1}^C \in$

$\mathbb{R}^{C\times d}$, we incorporate it into the global CD models to exploit multivariate correlations. Then, the final C-LoRA is given by:

$$\mathbf{Z}^{(0)} = \left[\mathtt{MLP}(\overline{\mathbf{X}};\theta)\|\mathbf{Z}_{\phi}^{(0)}\right] \in \mathbb{R}^{C\times(D+d)}, \tag{7}$$

where $[\cdot]$ is the concatenating operation, and $\mathbf{Z}^{(0)}$ is passed to the next modules. Note that C-LoRA achieves a balance between CD and CI models and efficiently integrates global-local components. It can adapt to individual channels with the specialized channel adaptation $\mathbf{z}_{c,\phi}^{(0)}$, and preserve multivariate interactions by the shared $\mathtt{MLP}(\overline{\mathbf{X}}_{:,c};\theta)$. The reduced hypothesis class is $\mathcal{H}_{\text{C-LoRA}} = \{\mathtt{MLP}(\cdot;\theta,\phi^{(c)})|\theta\in\Theta,\phi^{(c)}\in\mathbb{R}^{r\times D}\}$, which is more efficient.

4 EXPERIMENTS

4.1 Experimental Setup

To evaluate the effectiveness of C-LoRA, we conduct extensive experiments using popular LTSF benchmarks and various backbones.
Datasets. Following the same setting as in [9, 16], we consider 7 real-world time series datasets from various areas, including ETTh1, ETTm1, Weather, Electricity, Solar-Energy, PEMS04, and PEMS08.
Backbone architectures. We select a variety of neural forecasting architectures, including both CI and CD models. All of them are widely evaluated in previous work and have shown competitive performance. They are: Informer [23], Autoformer [17], FEDformer [24], FreTS [19], RMLP [7], TSMixer [2], and iTransformer [9].

4.2 Performance Improvement by C-LoRA

The results of long-term series forecasting are shown in Table 1. As indicated, C-LoRA can consistently improve the performance of different backbones, including both CI models and CM models. For datasets with more prominent *channel heterogeneity*, such as Solar and PeMS [10], the performance gains can be greater than 10%. For data with relatively small channels like ETT, we also observe improvements to some extent. Notably, models like RMLP can also benefit from C-LoRA even without explicit channel mixing modules.

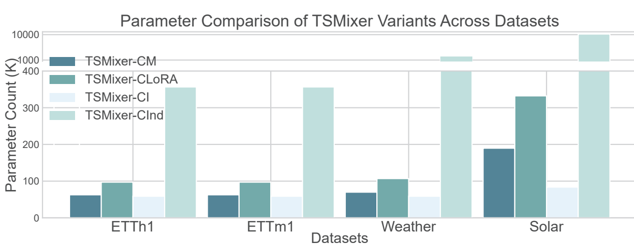

Figure 2: Parameter comparison of different strategies.

4.3 Parameter Efficiency

Due to the low-rank design, C-LoRA only introduces a few additional parameters to adapt the forecasting models. Fig. 2 compares it with both CI, CInd, and CD models. As can be seen, it significantly reduces the parameters of CInd models to consider individual treatment. In addition, as C-LoRA contains only a few extra computations, it adds very limited additional computational overhead.

4.4 Discussion and Analysis

Channel Mixing versus Channel Keeping. Fig. 3 examines the training and test errors of both CM and CI models under different look-back windows. First, both of them benefit from a longer look-back window and the use of C-LoRA. Second, the CD model has lower training errors, showing a larger capacity, however, can have larger generalization error than the CI model. Third, C-LoRA can narrow the performance gaps between training and test data.

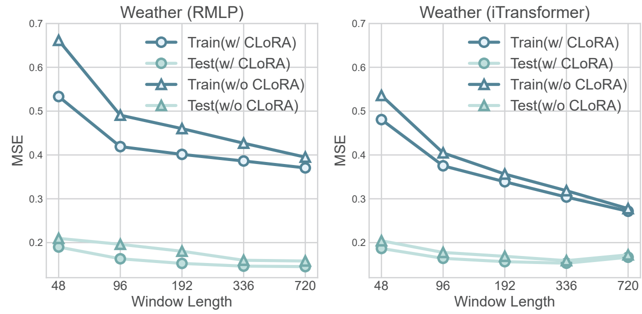

Figure 3: MSE under different lengths of look-back window.

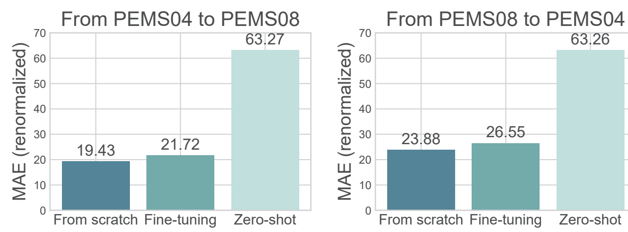

Figure 4: Transfer across datasets by fine-tuning C-LoRA.

Fine Tune C-LoRA. As shared patterns of time series can be captured by a CD model, a pretrained global model can be transferred to other datasets by efficient fine-tuning of C-LoRA. In Fig. 4, we compare the results of **all** parameters trained from scratch, **only** fine-tuning C-LoRA on the target set, and the zero-shot model. Surprisingly, it can achieve desirable performance by only fine-tuning C-LoRA, indicating the flexibility of our hybrid channel strategy.

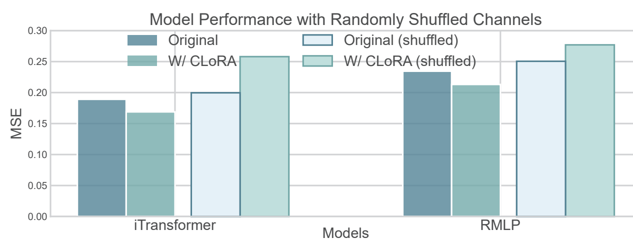

Figure 5: randomly shuffling the order of channels (Solar).

Channel Identity Permutation. By randomly shuffling the order of channels, we can evaluate the importance of channel identity. It is observed that both CI and CD models show a performance drop after shuffling the order of channels. In particular, models with C-LoRA have more pronounced error increases. This highlights that both CI and CD models can better preserve the channel identity information after being equipped with C-LoRA.
Impact of Rank Selection. Fig. 6 studies the impact of different rank values. For time series having more channels with redundancy, a lower intrinsic rank is more beneficial [12]. Conversely, a small data with heterogeneous channels prefer a large rank to ensure distinguishability. Both CI and CM methods show a similar trend.

Table 1: Results of the LTSF benchmarks. We report the forecast error of different models under different prediction lengths. The input sequence length is set to 96 for all methods. *IMP* shows the average percentage of MSE/MAE improvement of C-LoRA.

Models		iTransformer		w/ C-LoRA		TSMixer		w/ C-LoRA		RMLP		w/ C-LoRA		FreTS		w/ C-LoRA		FEDformer		w/ C-LoRA		Autoformer		w/ C-LoRA		Informer		w/ C-LoRA		IMP
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	%
ETTm1	96	0.345	0.376	**0.331**	**0.367**	0.332	0.370	**0.317**	**0.356**	0.337	0.374	**0.321**	**0.360**	0.340	0.376	**0.330**	**0.369**	0.379	0.419	**0.375**	**0.422**	**0.505**	**0.475**	0.512	0.480	0.672	0.571	**0.577**	**0.542**	3.23
	192	0.384	0.394	**0.373**	**0.390**	0.372	0.390	**0.358**	**0.377**	0.379	0.391	**0.363**	**0.380**	0.383	0.398	**0.375**	**0.398**	0.426	0.441	**0.411**	**0.435**	**0.553**	**0.496**	0.558	0.503	0.795	0.669	**0.720**	**0.634**	2.67
	336	0.418	0.415	**0.409**	**0.414**	0.405	0.411	**0.389**	**0.400**	0.412	0.412	**0.395**	**0.403**	0.420	0.425	**0.406**	**0.418**	0.445	0.459	**0.423**	**0.447**	0.621	0.537	**0.618**	**0.523**	1.212	0.871	**0.982**	**0.756**	4.51
	720	0.481	0.451	**0.479**	**0.449**	0.469	0.447	**0.455**	**0.436**	0.478	0.447	**0.462**	**0.440**	0.499	0.477	**0.479**	**0.462**	0.543	0.490	**0.509**	**0.488**	0.671	0.561	**0.592**	**0.520**	1.166	0.823	**1.121**	**0.794**	3.68
	Avg	0.407	0.409	**0.398**	**0.405**	0.395	0.405	**0.380**	**0.392**	0.402	0.406	**0.385**	**0.396**	0.411	0.419	**0.398**	**0.412**	0.448	0.452	**0.430**	**0.448**	0.588	0.517	**0.570**	**0.507**	0.961	0.734	**0.850**	**0.682**	3.59
ETTh1	96	**0.390**	**0.406**	**0.390**	0.407	0.398	0.412	**0.396**	**0.409**	0.405	0.413	**0.381**	**0.394**	**0.398**	0.410	0.402	**0.409**	**0.376**	0.419	**0.376**	**0.417**	**0.449**	**0.459**	0.453	0.469	0.912	0.717	**0.874**	**0.715**	0.90
	192	0.442	**0.436**	**0.440**	**0.436**	**0.451**	0.442	**0.451**	**0.440**	0.460	0.444	**0.442**	**0.427**	**0.453**	0.446	0.460	**0.444**	0.420	0.448	**0.419**	**0.443**	0.500	0.482	**0.490**	**0.477**	**1.037**	0.779	1.044	**0.775**	0.84
	336	0.485	0.459	**0.482**	**0.457**	**0.489**	0.461	0.493	**0.460**	0.505	0.466	**0.493**	**0.455**	**0.508**	0.480	0.509	**0.473**	0.459	**0.465**	**0.458**	0.466	0.521	0.496	**0.511**	**0.487**	**1.116**	**0.816**	1.146	0.843	0.30
	720	0.493	0.485	**0.486**	**0.477**	0.507	0.486	**0.498**	**0.480**	0.514	0.490	**0.493**	**0.475**	0.560	0.537	**0.547**	**0.515**	0.506	0.507	**0.472**	**0.486**	0.514	0.512	**0.503**	**0.508**	1.230	0.887	**1.170**	**0.869**	2.88
	Avg	0.453	0.447	**0.450**	**0.444**	0.461	0.450	**0.460**	**0.447**	0.471	0.453	**0.452**	**0.438**	0.480	0.468	0.480	**0.460**	0.440	0.460	**0.431**	**0.453**	0.496	0.487	**0.489**	**0.485**	1.074	0.800	**1.059**	0.801	1.28
Electricity	96	0.148	0.240	**0.139**	**0.234**	0.177	0.278	**0.155**	**0.256**	0.201	0.287	**0.168**	**0.258**	0.320	0.403	**0.165**	**0.262**	0.195	0.309	**0.193**	**0.307**	0.203	0.318	**0.193**	**0.307**	**0.329**	**0.407**	**0.329**	0.412	10.55
	192	0.162	0.253	**0.160**	**0.254**	0.193	0.293	**0.172**	**0.270**	0.209	0.297	**0.179**	**0.268**	0.325	0.406	**0.176**	**0.270**	**0.202**	**0.315**	0.203	0.316	0.225	0.334	**0.222**	**0.330**	**0.338**	**0.419**	0.347	0.430	8.53
	336	0.178	0.269	**0.171**	**0.266**	0.215	0.315	**0.191**	**0.289**	0.228	0.316	**0.196**	**0.285**	0.370	0.435	**0.195**	**0.289**	0.234	0.347	**0.230**	**0.343**	0.282	0.377	**0.266**	**0.370**	0.364	0.439	**0.352**	**0.434**	10.29
	720	0.225	0.317	**0.195**	**0.289**	0.260	0.352	**0.230**	**0.321**	0.273	0.350	**0.238**	**0.319**	0.416	0.474	**0.235**	**0.325**	0.261	**0.365**	0.262	**0.365**	0.314	**0.383**	**0.299**	0.389	0.397	0.460	**0.395**	**0.456**	10.24
	Avg	0.178	0.270	**0.166**	**0.261**	0.211	0.310	**0.187**	**0.284**	0.228	0.313	**0.195**	**0.283**	0.358	0.430	**0.193**	**0.287**	0.223	0.334	**0.222**	**0.333**	0.256	0.353	**0.245**	**0.349**	0.357	0.431	**0.356**	0.433	9.95
Weather	96	0.174	0.214	**0.164**	**0.209**	0.181	0.228	**0.158**	**0.206**	0.196	0.235	**0.163**	**0.208**	0.186	0.241	**0.165**	**0.228**	0.220	0.300	**0.218**	**0.299**	0.266	0.336	**0.234**	**0.312**	0.300	0.384	**0.265**	**0.348**	8.35
	192	0.221	0.254	**0.209**	**0.251**	0.227	0.263	**0.207**	**0.249**	0.240	0.271	**0.209**	**0.249**	0.222	**0.273**	**0.210**	0.274	**0.278**	**0.344**	0.282	0.350	0.307	0.367	**0.282**	**0.344**	0.598	0.544	**0.381**	**0.427**	8.27
	336	0.278	0.296	**0.268**	**0.294**	0.280	0.300	**0.266**	**0.292**	0.291	0.307	**0.264**	**0.289**	0.272	0.316	**0.264**	**0.317**	**0.339**	**0.382**	0.349	0.390	0.359	**0.395**	**0.357**	**0.395**	0.578	0.523	**0.515**	**0.511**	2.74
	720	0.358	0.349	**0.349**	**0.346**	0.353	0.347	**0.348**	**0.345**	0.363	0.353	**0.342**	**0.340**	0.350	0.381	**0.343**	**0.370**	0.409	0.438	0.411	**0.420**	0.419	0.428	**0.415**	**0.424**	1.059	0.741	**0.792**	**0.651**	4.47
	Avg	0.258	0.278	**0.248**	**0.275**	0.260	0.285	**0.245**	**0.273**	0.273	0.292	**0.245**	**0.272**	0.258	0.303	**0.246**	**0.297**	0.312	0.366	0.315	**0.365**	0.338	0.382	**0.322**	**0.369**	0.634	0.548	**0.488**	**0.484**	5.76
Solar	96	0.203	0.237	**0.175**	**0.219**	0.222	0.281	**0.182**	**0.272**	0.233	0.296	**0.213**	**0.272**	0.237	0.300	**0.231**	**0.295**	0.242	0.342	**0.226**	**0.319**	0.884	0.711	**0.603**	**0.545**	0.236	0.279	**0.214**	**0.245**	10.96
	192	0.233	0.261	**0.211**	**0.259**	0.261	0.301	**0.207**	**0.275**	0.260	0.316	**0.234**	**0.292**	0.265	0.321	**0.261**	**0.318**	0.285	0.380	**0.245**	**0.366**	0.834	0.692	**0.682**	**0.563**	0.227	0.287	0.241	0.290	7.64
	336	0.248	0.273	**0.222**	**0.261**	0.271	0.299	**0.212**	**0.272**	0.276	0.323	**0.247**	**0.301**	0.283	0.330	**0.277**	**0.325**	0.282	0.376	**0.246**	**0.350**	0.941	0.723	**0.739**	**0.588**	0.262	0.310	**0.246**	**0.307**	9.54
	720	0.249	0.275	**0.203**	**0.263**	0.267	0.293	**0.201**	**0.262**	0.273	0.316	**0.244**	**0.291**	0.286	0.326	**0.281**	**0.322**	0.357	0.427	**0.304**	**0.410**	0.882	0.717	**0.801**	**0.642**	0.329	0.355	**0.279**	**0.329**	10.05
	Avg	0.233	0.262	**0.203**	**0.251**	0.255	0.294	**0.201**	**0.270**	0.261	0.313	**0.235**	**0.289**	0.268	0.319	**0.263**	**0.315**	0.292	0.381	**0.255**	**0.361**	0.885	0.711	**0.706**	**0.585**	0.264	0.308	**0.245**	**0.293**	9.65
PEMS04	96	0.159	0.272	**0.124**	**0.237**	0.166	0.285	**0.111**	**0.220**	0.273	0.372	**0.180**	**0.292**	0.274	0.372	**0.211**	**0.311**	0.220	0.338	**0.219**	**0.336**	0.553	0.583	**0.398**	**0.471**	**0.123**	**0.236**	0.125	**0.236**	16.60
	192	0.182	0.290	**0.162**	**0.266**	0.196	0.166	**0.127**	**0.235**	0.308	0.395	**0.224**	**0.320**	0.307	0.392	**0.248**	**0.342**	0.313	0.419	**0.310**	**0.418**	0.938	0.772	**0.502**	**0.546**	0.142	0.252	**0.136**	**0.246**	12.48
	336	0.186	0.291	**0.166**	**0.268**	0.204	0.312	**0.134**	**0.242**	0.283	0.373	**0.224**	**0.322**	0.285	0.375	**0.247**	**0.346**	**0.231**	**0.339**	0.232	**0.339**	0.953	0.772	**0.672**	**0.645**	0.155	0.263	**0.148**	**0.256**	13.12
	720	0.242	0.332	**0.203**	**0.298**	0.238	0.345	**0.148**	**0.259**	0.328	0.407	**0.258**	**0.352**	0.329	0.406	**0.286**	**0.376**	0.643	0.592	**0.605**	**0.577**	1.120	0.823	**0.879**	**0.747**	0.192	0.297	**0.162**	**0.272**	14.83
	Avg	0.192	0.296	**0.164**	**0.267**	0.201	0.277	**0.130**	**0.239**	0.298	0.387	**0.222**	**0.322**	0.299	0.386	**0.248**	**0.344**	0.352	0.422	**0.342**	**0.418**	0.891	0.738	**0.613**	**0.602**	0.153	0.262	**0.143**	**0.253**	14.86
PEMS08	96	0.169	0.276	**0.109**	**0.216**	0.252	0.355	**0.124**	**0.239**	0.284	0.375	**0.198**	**0.300**	0.285	0.380	**0.220**	**0.328**	**0.221**	**0.325**	0.234	0.329	0.613	0.596	**0.444**	**0.508**	0.171	0.283	**0.165**	**0.274**	19.24
	192	0.188	0.288	**0.137**	**0.239**	0.322	0.385	**0.151**	**0.256**	0.336	0.409	**0.222**	**0.310**	0.335	0.409	**0.280**	**0.371**	**0.332**	**0.412**	0.340	0.420	1.111	0.829	**0.616**	**0.596**	**0.202**	0.304	**0.202**	**0.298**	20.35
	336	0.196	0.289	**0.148**	**0.243**	0.326	0.374	**0.170**	**0.265**	0.327	0.394	**0.266**	**0.337**	0.328	0.397	**0.284**	**0.363**	**0.247**	0.322	**0.247**	**0.321**	1.136	0.836	**0.716**	**0.648**	0.224	0.309	**0.210**	**0.299**	17.27
	720	0.235	0.320	**0.186**	**0.277**	0.388	0.422	**0.225**	**0.308**	0.372	0.430	**0.303**	**0.372**	0.373	0.431	**0.327**	**0.396**	0.564	0.556	**0.546**	**0.546**	1.409	0.948	**0.901**	**0.733**	0.247	0.324	**0.238**	**0.316**	16.12
	Avg	0.197	0.293	**0.145**	**0.244**	0.322	0.384	**0.168**	**0.267**	0.330	0.402	**0.247**	**0.330**	0.330	0.404	**0.278**	**0.365**	0.341	0.404	**0.342**	**0.404**	1.067	0.802	**0.669**	**0.621**	0.211	0.305	**0.204**	**0.297**	18.30

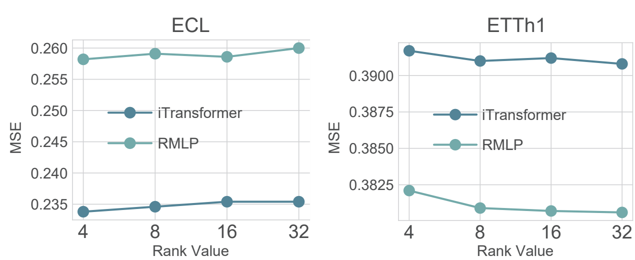

Figure 6: Model performance under different rank values.

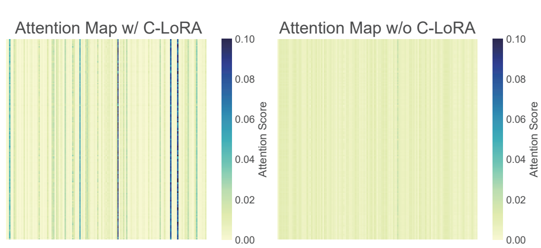

Figure 7: Attention scores w/ and w/o C-LoRA.

Case Study: Enhanced Attention Maps. We visualize the attention maps of iTransformer of the same input with and without C-LoRA. Clearly, C-LoRA can help generate a attention map with more prominent patterns, focusing on a few critical channels. This indicates the role of preserving channel identity with C-LoRA.

5 CONCLUSION

We introduce C-LoRA, a novel channel strategy to combine the strengths of both CI and CD strategies for long-term series forecasting. C-LoRA adapts to each channel with both channel-specific low-rank matrices and local patterns, bypassing the need for individual models. Potential cross-channel dependencies are preserved by a globally shared predictor. Extensive results indicate that it can consistently improve the performance of various backbones by a large margin. It is also efficient, flexible to fine-tune, and beneficial for the exploitation of channel identity. Future work can evaluate its scalability in larger datasets, such as stock [14] and traffic [8].

ACKNOWLEDGMENTS

The work was supported by research grants from the NSFC (52125208).

REFERENCES

[1] Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassiulas, Jure Leskovec, and Rex Ying. 2024. From Similarity to Superiority: Channel Clustering for Time Series Forecasting. *arXiv preprint arXiv:2404.01340* (2024).

[2] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. 2023. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053* (2023).

[3] Andrea Cini, Ivan Marisca, Daniele Zambon, and Cesare Alippi. 2024. Taming local effects in graph-based spatiotemporal forecasting. *Advances in Neural Information Processing Systems* 36 (2024).

[4] Lu Han, Han-Jia Ye, and De-Chuan Zhan. 2024. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[5] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[6] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

[7] Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. 2023. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721* (2023).

[8] Xu Liu, Yutong Xia, Yuxuan Liang, Junfeng Hu, Yiwei Wang, Lei Bai, Chao Huang, Zhenguang Liu, Bryan Hooi, and Roger Zimmermann. 2024. Largest: A benchmark dataset for large-scale traffic forecasting. *Advances in Neural Information Processing Systems* 36 (2024).

[9] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625* (2023).

[10] Tong Nie, Guoyang Qin, Wei Ma, Yuewen Mei, and Jian Sun. 2023. ImputeFormer: Low rankness-induced transformers for generalizable spatiotemporal imputation. *arXiv: 2312.01728* (2023).

[11] Tong Nie, Guoyang Qin, Lijun Sun, Wei Ma, Yu Mei, and Jian Sun. 2023. Contextualizing MLP-Mixers Spatiotemporally for Urban Data Forecast at Scale. *arXiv preprint arXiv:2307.01482* (2023).

[12] Tong Nie, Guoyang Qin, Yunpeng Wang, and Jian Sun. 2023. Correlating sparse sensing for large-scale traffic speed estimation: A Laplacian-enhanced low-rank tensor kriging approach. *Transportation research part C: emerging technologies* 152 (2023), 104190.

[13] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022).

[14] Guowei Song, Tianlong Zhao, Suwei Wang, Hua Wang, and Xuemei Li. 2023. Stock ranking prediction using a graph aggregation network based on stock price and stock relationship information. *Information Sciences* 643 (2023), 119236.

[15] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and JUN ZHOU. 2023. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *The Twelfth International Conference on Learning Representations*.

[16] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2022. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The eleventh international conference on learning representations*.

[17] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems* 34 (2021), 22419–22430.

[18] Jingyun Xiao, Ran Liu, and Eva L Dyer. 2023. GAFormer: Enhancing Timeseries Transformers Through Group-Aware Embeddings. In *The Twelfth International Conference on Learning Representations*.

[19] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. 2024. Frequency-domain MLPs are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems* 36 (2024).

[20] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting?. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 11121–11128.

[21] Yunhao Zhang and Junchi Yan. 2022. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

[22] Lifan Zhao and Yanyan Shen. 2024. Rethinking Channel Dependence for Multivariate Time Series Forecasting: Learning from Leading Indicators. *arXiv preprint arXiv:2401.17548* (2024).

[23] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11106–11115.

[24] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*. PMLR, 27268–27286.