Partial Channel Dependence with Channel Masks for Time Series Foundation Models

Seunghan Lee, Taeyoung Park,* Kibok Lee*
Department of Statistics and Data Science, Yonsei University
{seunghan9613,tpark,kibok}@yonsei.ac.kr

Abstract

While advances in foundation models have extended to the time series domain, they have primarily focused on designing model architectures to address external heterogeneity between datasets, e.g., varying numbers of channels, often overlooking internal heterogeneity, e.g., varying channel dependencies. In this work, we introduce the concept of partial channel dependence (PCD), which enables a more sophisticated adjustment of channel dependencies based on dataset-specific information. To achieve PCD, we propose a channel mask that captures the relationships between channels within a dataset using two key components: 1) a correlation matrix that encodes relative dependencies between channels, and 2) domain parameters that learn the absolute dependencies specific to each dataset, refining the correlation matrix. We validate the effectiveness of our method across various tasks, including forecasting, classification, imputation, and anomaly detection.

1 Introduction

Foundation models (FMs) have emerged in various domains [29, 26, 14], including the time series (TS) domain [9, 19]. These models are pretrained on diverse datasets and are designed to solve multiple tasks using a single model. Directly applying FMs to TS is, however, challenging due to the *heterogeneity* among TS datasets [9, 31], so that various time series foundation models (TSFMs) have been proposed. While these approaches mainly focus on *explicit heterogeneity*, where datasets differ in observable characteristics such as varying sequence lengths and number of channels in TS, they tend to overlook *implicit heterogeneity*, which involves unobservable factors such as differences in inter-channel dependencies. Furthermore, these methods address heterogeneity by modifying the model architecture, often overlooking the inherent characteristics of the dataset.

In this paper, we consider the implicit heterogeneity among TS datasets when building a TSFM, specifically the varying channel dependencies (CD) across datasets, as opposed to prior TSFMs that mainly address the explicit heterogeneity and TS forecasting models that focus solely on adjusting the model architecture to capture CD. We argue that addressing this implicit heterogeneity is crucial for TSFMs because assuming a uniform model across all datasets can be problematic due to the varying CD across datasets, as shown in Figure 1. To this end, we introduce the concept of *partial channel dependence* (PCD) which adjusts the CD estimated by the model by leveraging the characteristics of the dataset, Specifically, we propose a *channel mask* (CM) that adjusts the dependencies between channels to achieve PCD. A CM consists of 1) a **correlation matrix** to encode relative dependencies between channels and 2) **domain parameters** that learn the absolute dependencies specific to each dataset to refine the matrix. The proposed CM, constructed using dataset-specific information, is multiplied to the (channel-wise) attention matrix (i.e., CD estimated by the model). The main contributions are summarized as follows:

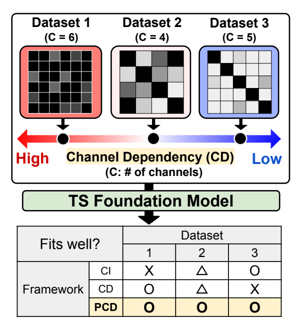

Figure 1: Varying CD.

- We introduce the concept of partial channel dependence (PCD), where the channel dependence (CD) captured by the model is partially adjusted based on the characteristics of the TS dataset.

*Equal advising.

NeurIPS 2024 Workshop on Time Series in the Age of Large Models.

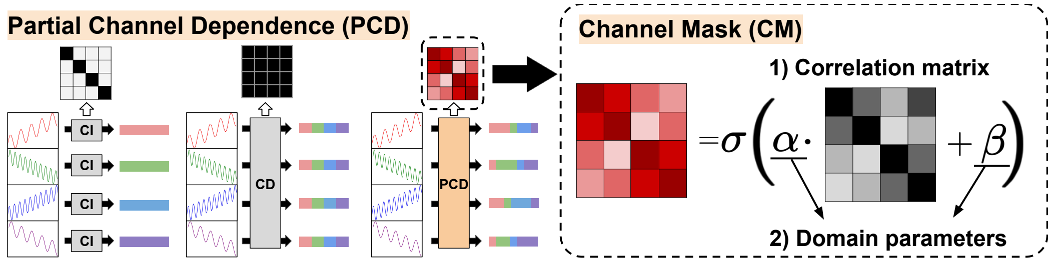

Figure 2: **CM for PCD.** To achieve PCD, we propose a CM, which consists of a correlation matrix between channels and domain parameters that refine the matrix based on the dataset.

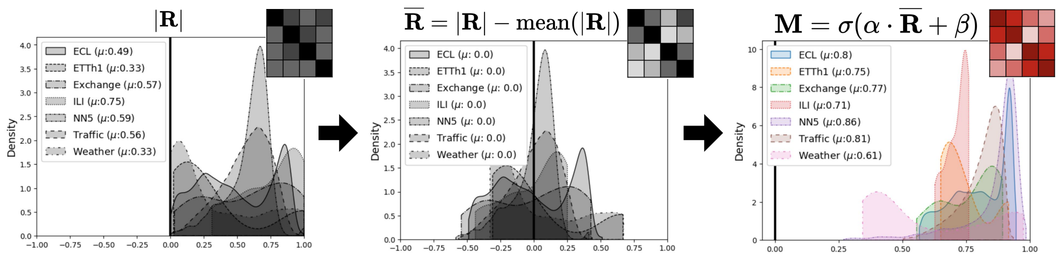

Figure 3: **Domain parameters to adjust correlation matrix.** As correlation is a relative measure depending on the dataset, we refine the correlation matrix using the domain parameters.

- We propose a channel mask (CM) to capture relative dependencies between channels and absolute dependencies specific to each dataset using correlation matrix and domain parameters, respectively.
- We present extensive experiments with both single-task models and multi-task FMs across four different tasks under various settings, demonstrating consistent performance gains.

2 Methodology

In this section, we introduce a CM, which employs a correlation matrix to capture relative dependencies between channels and adjusts it with domain parameters to learn absolute dependencies specific to each dataset. We also introduce a channel dependence ratio (CD ratio), which uses a CM to quantify the degree of CD for each dataset. The overall framework of a CM is illustrated in Figure 2.

2.1 Components of Channel Mask

As shown in Figure 2, a CM consists of two components: 1) correlation matrix ($\mathbf{R}$) between channels, and 2) domain parameters (α and β), which scale and shift the matrix according to the dataset's characteristics, along with a sigmoid function to normalize the values between 0 and 1.

Correlation matrix. Correlation measures the relationships between channels and has been used in previous works to analyze CD [35, 38]. Building on these approaches, we employ a correlation matrix ($\mathbf{R}$) between channels to create a CM. However, high correlation does not always indicate a strong positive relationship, as the values range from -1 to 1, with strong negative relationships near -1. To address this issue, we use the absolute value of the matrix $|\mathbf{R}|$.

Domain parameters. We argue that $|\mathbf{R}|$ alone might be insufficient for modeling a CM for the following reasons: First, correlation is a relative measure that depends on the dataset. As shown in the first panel of Figure 3, different datasets exhibit different distributions of the elements of $|\mathbf{R}|$. To align these differences, we normalize $|\mathbf{R}|$ by subtracting the mean value, resulting in $\bar{\mathbf{R}}$, as shown in the second panel of Figure 3. Second, the relationship between correlation and CD may vary across datasets (i.e., the same correlation can correspond to different levels of CD depending on the dataset). To deal with this discrepancy among datasets, we introduce two learnable domain parameters, α and β, which scale and shift $|\mathbf{R}|$, respectively, as shown in the third panel of Figure 3. Using these parameters along with a sigmoid function, we model a CM for achieving PCD as $\mathbf{M} = \sigma(\alpha \cdot \bar{\mathbf{R}} + \beta)$.

2.2 Channel Mask with Attention Matrix

The proposed CM adjusts the CD estimated by the model by performing element-wise multiplication with the attention matrix of Transformers, with the general adjustment modeled by $\mathbf{A}$:

$$\mathrm{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{Softmax}\left(\mathbf{A} \odot \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\cdot \mathbf{V}, \text{ where } \mathbf{A} = \begin{cases} \mathbf{I}_{C\times C} & \text{if CI,} \\ \mathbf{1}_{C\times C} & \text{if CD,} \\ \mathbf{M} = \sigma(\alpha \cdot \bar{\mathbf{R}} + \beta) & \text{if PCD,} \end{cases} \quad (1)$$

and C is the number of channels. Note that Equation 1 incorporates both CI and CD frameworks within a single expression: As shown in Figure 2, $\mathbf{A}$ is the identity matrix ($\mathbf{I}_{C\times C}$) in the CI framework,

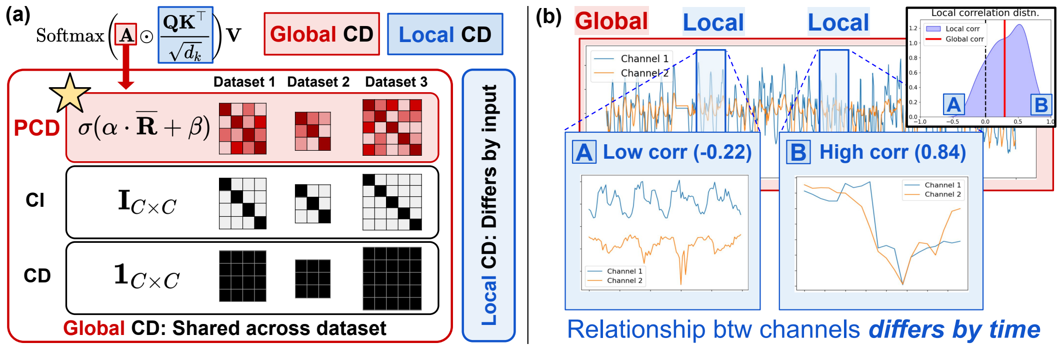

Figure 4: **Global and local CD.** (a) shows a CM and an attention matrix, which capture the global and local dependencies between channels, respectively. (b) illustrates the global and local correlations between two channels of ETTh1 [39], revealing that local correlations can vary by input TS.

while $\mathbf{A}$ is a matrix of ones ($\mathbf{1}_{C\times C}$) in the CD framework. In contrast, our PCD framework represents $\mathbf{A}$ as $\mathbf{M} = \sigma(\alpha \cdot \bar{\mathbf{R}} + \beta)$, enabling a more refined adjustment of CD tailored to the dataset.

Global and local CD. As a correlation matrix is calculated based on the entire TS dataset, a CM captures the global CD, which represents the CD shared across all time steps. This complements the local CD captured by an attention matrix, which represents the CD that varies by input time step. As shown in Figure 4(a), our PCD framework captures both global and local CDs through the element-wise multiplication of a CM and an attention matrix ($\mathbf{QK}^\top/\sqrt{d_k}$). Furthermore, Figure 4(b), which illustrates two channels of ETTh1 [39], shows that the dependency can differ across time steps even within the same dataset, underscoring the need to capture both global and local CDs. Further analysis on the necessity of capturing both CDs is discussed in Table 6.

2.3 Channel Dependence Ratio

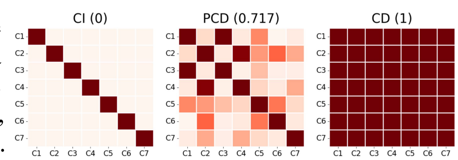

Figure 5: CD ratio of CI/PCD/CD.

To quantify the degree of CD for each dataset, we propose to measure the *channel dependence ratio* (CD ratio), a metric based on a CM. The CD ratio of $\mathbf{M}$, denoted as $r(\mathbf{M})$, is the average of the off-diagonal elements of $\mathbf{M}$, excluding the autocorrelations of their respective channels. This metric yields a value of 0 for CI cases and 1 for CD cases, with higher values indicating a greater preference for interaction between channels. Figure 5 shows the visualization of $\mathbf{M}$ and its corresponding CD ratio for ETTh1 [39], with a ratio of 0.717 for PCD. We find that $\mathbf{M}$ effectively captures the degree of CD for each dataset, as datasets with higher $r(\mathbf{M})$ tend to have greater performance gains with CD architecture compared to CI architecture, as illustrated in Figure G.2.

3 Experiments

We demonstrate the effectiveness of our method by applying it to both single-task and multi-task models in either supervised (SL) or self-supervised (SSL) settings, with iTransformer (iTrans.) [18] for single-task SL, TimeSiam [6] for single-task SSL, and UniTS [7] for multi-task SSL. As shown in Table 1, we perform four different tasks: forecasting (FCST), classification (CLS), imputation (IMP), and anomaly detection (AD), across various data ratios including few-shot and zero-shot settings. As evaluation metrics, we use the mean squared error (MSE) and mean absolute error (MAE) for FCST and IMP, accuracy (Acc.) for CLS, and F_1 score for AD. Dataset statistics and implementation details can be found in Appendix A and B, respectively.

Model			TS downstream tasks				Data %	Section	
			FCST	CLS	IMP	AD		Summary	Full
Single-task	SL	iTransformer	✓	-	-	-	Full	Table 3	Appendix D
	SSL	TimeSiam	✓	-	-	-		-	Appendix F
Multi-task (FM)	SSL	UniTS	✓	✓	-	-	Full	Table 4	Appendix E.1
			✓	✓	✓	✓	Few-shot	Appendix E.2	Appendix E.3
			✓	-	-	-	Zero-shot	-	Appendix E.4

Table 1: Summary of experiments.

20 Tasks		Shared (1 model)								Task-specific (20 models)							
		UniTS + CM				UniTS				iTransformer		TimesNet		PatchTST		GPT4TS	
		Sup.		PT		Sup.		PT		Sup.						FT	
Dataset	H	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NN5	112	0.641	0.568	**0.586**	**0.536**	0.635	0.556	0.611	0.552	0.623	0.554	0.629	0.541	0.634	0.568	0.623	0.545
ECL	96	0.176	0.278	**0.168**	**0.272**	0.172	0.273	0.174	0.277	0.204	0.288	0.184	0.289	0.212	0.299	0.198	0.285
	192	0.188	0.287	**0.184**	0.286	0.185	**0.284**	0.189	0.289	0.208	0.294	0.204	0.307	0.213	0.303	0.200	0.288
	336	0.199	**0.295**	0.199	0.301	**0.196**	0.297	0.205	0.304	0.224	0.310	0.217	0.320	0.228	0.317	0.214	0.302
	720	**0.230**	**0.321**	0.231	0.326	0.238	**0.321**	0.251	0.340	0.265	0.341	0.284	0.363	0.270	0.348	0.254	0.333
ETTh1	96	0.388	0.405	0.389	0.408	0.390	0.408	0.390	0.411	**0.382**	**0.399**	0.478	0.448	0.389	0.400	0.396	0.413
	192	0.438	0.436	0.432	0.432	**0.428**	0.432	0.432	0.439	0.431	**0.426**	0.561	0.504	0.440	0.43	0.458	0.448
	336	0.478	0.455	0.475	0.451	**0.462**	0.451	0.480	0.460	0.476	**0.449**	0.612	0.537	0.482	0.453	0.508	0.472
	720	**0.483**	**0.472**	0.515	0.492	0.489	0.476	0.532	0.500	0.495	0.487	0.601	0.541	0.486	0.479	0.546	0.503
Exchange	192	0.231	0.340	0.210	0.330	0.239	0.342	0.221	0.337	**0.175**	**0.297**	0.259	0.370	0.178	0.301	0.177	0.300
	336	0.431	0.472	0.387	0.451	0.479	0.486	0.387	0.453	**0.322**	**0.409**	0.478	0.501	0.328	0.415	0.326	0.414
ILI	60	2.02	**0.885**	2.15	0.923	2.48	0.944	2.45	0.994	**1.99**	0.905	2.37	0.966	2.31	0.970	1.90	0.868
Traffic	96	0.486	**0.322**	**0.483**	0.324	0.496	0.325	0.502	0.330	0.606	0.389	0.611	0.336	0.643	0.405	0.524	0.351
	192	**0.492**	**0.325**	0.500	0.330	0.497	0.327	0.523	0.331	0.592	0.382	0.643	0.352	0.603	0.387	0.519	0.346
	336	**0.506**	0.331	0.520	0.337	0.509	**0.328**	0.552	0.338	0.600	0.384	0.662	0.363	0.612	0.389	0.530	0.350
	720	**0.523**	**0.340**	0.575	0.362	0.525	0.350	0.626	0.369	0.633	0.401	0.678	0.365	0.652	0.406	0.562	0.366
Weather	96	0.165	**0.211**	0.166	0.219	**0.161**	**0.211**	0.175	0.214	0.193	0.232	0.169	0.220	0.194	0.233	0.182	0.222
	192	**0.210**	**0.254**	0.216	0.261	0.212	0.255	0.226	0.266	0.238	0.269	0.223	0.264	0.238	0.268	0.228	0.261
	336	**0.266**	**0.294**	0.273	0.300	**0.266**	0.295	0.280	0.303	0.291	0.306	0.279	0.302	0.290	0.304	0.282	0.299
	720	**0.342**	**0.343**	0.350	0.349	0.343	0.344	0.352	0.350	0.365	0.354	0.359	0.355	0.363	0.35	0.359	0.349
Best Count (/20)		8	11	4	2	5	4	0	0	4	5	0	0	0	0	-	-
Average		**0.445**	**0.382**	0.452	0.384	0.469	0.386	0.478	0.393	0.466	0.394	0.525	0.412	0.488	0.401	0.449	0.386

Table 2: Results of multi-task forecasting.

	ETTh1	ETTh2	ETTm1	ETTm2	PEMS03	PEMS04	PEMS07	PEMS08	Exchange	Weather	Solar	ECL	Traffic	Avg.
iTrans.	0.457	0.384	0.408	0.293	0.142	0.121	0.102	0.254	0.368	0.260	0.234	0.179	0.428	0.279
+ CM	**0.444**	**0.383**	**0.398**	**0.289**	**0.124**	**0.098**	**0.082**	**0.152**	**0.363**	**0.250**	**0.228**	**0.168**	**0.422**	**0.261**
Impr.	**2.8%**	**0.3%**	**2.5%**	**1.4%**	**12.7%**	**19.0%**	**19.6%**	**40.2%**	**1.4%**	**3.8%**	**2.6%**	**6.1%**	**1.4%**	**6.3%**

Table 3: Results of single-task forecasting.

		UniTS	+ CM	Impr.
FCST (MSE)	Sup.	0.469	**0.445**	**5.1%**
	PT	0.478	**0.452**	**5.4%**
CLS (Acc.)	Sup.	80.6	**82.0**	**1.7%**
	PT	75.1	**78.3**	**4.3%**

Table 4: 20 FCST and 18 CLS tasks.

Components		M	FCST (20)		CLS (18)
Corr.	Domain		MSE	MAE	Acc.
		$\mathbf{1}$	0.478	0.393	75.1%
✓		$\lvert\mathbf{R}\rvert$	0.474	0.390	78.8%
✓		$\bar{\mathbf{R}}$	0.471	0.388	78.1%
	✓	$\sigma\left(\alpha \cdot \mathbf{I} + \beta\right)$	0.497	0.406	76.2%
✓	✓	$\sigma\left(\alpha \cdot \bar{\mathbf{R}} + \beta\right)$	**0.452**	**0.384**	**80.6%**

Table 5: Ablation study of CM.

Global	Local	ETTh1	ETTh2	ETTm1	ETTm2	PEMS03	PEMS04	PEMS07	PEMS08	Exchange	Weather	Solar	ECL	Traffic	Avg.
✓		0.466	**0.383**	**0.398**	**0.289**	0.206	0.116	0.101	0.162	**0.363**	0.259	0.233	0.176	0.429	0.275
	✓	0.457	0.384	0.408	0.293	0.142	0.121	0.102	0.254	0.368	0.260	0.234	0.179	0.428	0.279
✓	✓	**0.444**	**0.383**	**0.398**	**0.289**	**0.124**	**0.098**	**0.082**	**0.152**	**0.363**	**0.250**	**0.228**	**0.168**	**0.422**	**0.261**

Table 6: Effect of capturing global and local CD.

Application to iTransformer. To show the effectiveness of CMs, we apply CMs to iTransformer to solve TS forecasting tasks on 13 different datasets. Table 3 shows the result with the average MSE and MAE across four different horizons, showing consistent improvement across all datasets.

Application to UniTS. To validate the effectiveness of our method on TSFM, we apply CMs to UniTS which solves diverse tasks without the need for fine-tuning, relying solely on prompt-tuning. Table 4 shows the brief results of 20 FCST tasks and 18 CLS tasks under both supervised (Sup.) and prompt-tuning (PT) settings, where full results are shown in Table 2 and Appendix E.1, respectively. The results indicate that applying our method improves performance in all 20 FCST tasks compared to not using our method, and enhances 13 CLS tasks for UniTS. Additionally, compared to GPT4TS [40], which is a TSFM that reprograms the pretrained GPT-2 model [25], our method achieves superior performance with less than 1% of the parameters (164.5M vs. 1.57M).

Effect of CM. To demonstrate the effectiveness of CMs, we conduct ablation studies based on the use of correlation matrix and domain parameters with UniTS under the prompt-tuning setting. To isolate the effect of using only the domain parameters, we replace $\bar{\mathbf{R}}$ with the identity matrix $\mathbf{I}$. Table 5 shows the result, indicating that incorporating both components leads to the best performance.

Global & local CD. To demonstrate the effect of attention matrices capturing the local CD of the input TS and CMs capturing the global CD of the entire TS, we conduct an ablation study, as shown in Table 6. Specifically, to observe the local, global, and combined effects, we use the attention weights $\mathbf{W}$ in $\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Softmax}\left(\mathbf{W}\right) \cdot \mathbf{V}$ in the following manner: $\mathbf{Q}\mathbf{K}^\top/\sqrt{d_k}$ for local CD, $\mathbf{M}$ for global CD, and $\mathbf{M} \odot \mathbf{Q}\mathbf{K}^\top/\sqrt{d_k}$ for both. The results show the average MSE for four different horizons, indicating that using both components yields the best results.

References

[1] Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2485–2494, 2021.

[2] Md Atik Ahamed and Qiang Cheng. Timemachine: A time series is worth 4 mambas for long-term forecasting. *arXiv preprint arXiv:2403.09898*, 2024.

[3] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[4] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting. *TMLR*, 2023.

[5] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[6] Jiaxiang Dong, Haixu Wu, Yuxuan Wang, Yunzhong Qiu, Li Zhang, Jianmin Wang, and Mingsheng Long. Timesiam: A pre-training framework for siamese time-series modeling. In *ICML*, 2024.

[7] Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: Building a unified time series model. In *NeurIPS*, 2024.

[8] Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.

[9] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *ICML*, 2024.

[10] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395, 2018.

[11] Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

[12] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. In *ICLR*, 2024.

[13] D Kinga, Jimmy Ba Adam, et al. A method for stochastic optimization. In *ICLR*, volume 5, page 6. San Diego, California;, 2015.

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023.

[15] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[16] Seunghan Lee, Taeyoung Park, and Kibok Lee. Learning to embed time series patches independently. In *ICLR*, 2024.

[17] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. In *NeurIPS*, 2022.

[18] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2024.

[19] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *ICML*, 2024.

[20] Aditya P Mathur and Nils Ole Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pages 31–36. IEEE, 2016.

[21] AI McLeod and Hyukjun Gweon. Optimal deseasonalization for monthly and daily geophysical time series. *Journal of Environmental statistics*, 4(11):1–11, 2013.

[22] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, pages 1–74, 2024.

[23] Yushan Nie, Nam H Nguyen, Pattarawat Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.

[24] NREL. Solar power data for integration studies. `https://www.nrel.gov/grid/solar-power-data.html`, 2006.

[25] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[27] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2828–2837, 2019.

[28] Souhaib Ben Taieb, Gianluca Bontempi, Amir F Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert systems with applications*, 39(8):7067–7083, 2012.

[29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[31] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *ICML*, 2024.

[32] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023.

[33] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, 2021.

[34] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *IJCAI*, 2019.

[35] Yingnan Yang, Qingling Zhu, and Jianyong Chen. Vcformer: Variable correlation transformer with inherent lagged correlation for multivariate time series forecasting. *arXiv preprint arXiv:2405.11470*, 2024.

[36] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI*, 2023.

[37] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *ICLR*, 2023.

[38] Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In *ICLR*, 2024.

[39] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, 2021.

[40] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. In *NeurIPS*, 2023.

A Dataset Description

A.1 Dataset for Single-task Model: iTransformer

For TS forecasting in a single-task setting, we evaluate the effectiveness of our proposed method using 13 datasets, with their statistics described in Table A.1. We adhere to the same data processing and train-validation-test split protocol as iTransformer [18], ensuring that the training, validation, and test sets are separated in chronological order. The input length is consistently set to 96 across all datasets. Note that N and C denote the size of the dataset and number of channels in a dataset, respectively.

Dataset	C	Prediction Length	$(N_{\text{train}}, N_{\text{val}}, N_{\text{test}})$
ETTh1 [39]	7	{96, 192, 336, 720}	(8545, 2881, 2881)
ETTh2 [39]	7	{96, 192, 336, 720}	(8545, 2881, 2881)
ETTm1 [39]	7	{96, 192, 336, 720}	(34465, 11521, 11521)
ETTm2 [39]	7	{96, 192, 336, 720}	(34465, 11521, 11521)
Exchange [33]	8	{96, 192, 336, 720}	(5120, 665, 1422)
Weather [33]	21	{96, 192, 336, 720}	(36792, 5271, 10540)
ECL [33]	321	{96, 192, 336, 720}	(18317, 2633, 5261)
Traffic [33]	862	{96, 192, 336, 720}	(12185, 1757, 3509)
Solar-Energy [15]	137	{96, 192, 336, 720}	(36601, 5161, 10417)
PEMS03 [17]	358	{12, 24, 48, 96}	(15617, 5135, 5135)
PEMS04 [17]	307	{12, 24, 48, 96}	(10172, 3375, 3375)
PEMS07 [17]	883	{12, 24, 48, 96}	(16911, 5622, 5622)
PEMS08 [17]	170	{12, 24, 48, 96}	(10690, 3548, 3548)

Table A.1: Single-task forecasting datasets.

A.2 Dataset for Multi-task Model: UniTS

The datasets used in the experiment are aggregated from the Monash Forecasting Repository [8], the Time Series Classification Website [22], and the Time Series Library [32]. The combined training set includes more than 35 million time steps and over 6,000 variables (channels). Note that N, L, C denote the training size, input length, and number of channels in a dataset, respectively.

A.2.1 Multi-task Learning

For TS forecasting and classification in a multi-task setting, we evaluate the effectiveness of our proposed method using 20 datasets for forecasting and 18 datasets for classification. The statistics of these datasets are summarized in Table A.2 and A.3, respectively.

Category	Dataset	Prediction Length	N	L	C
Finance	NN5 [28]	112	409	112	111
	Exchange [33]	192 336	5024 4880	96	8
Electricity	ECL [33]	96 192 336 720	18221 18125 17981 17597	96	321
	ETTh1 [39]	96 192 336 720	8449 8353 8209 7825	96	7
Illness	ILI [33]	60	581	36	7
Traffic	Traffic [33]	96 192 336 720	12089 11993 11849 11465	96	862
Weather	Weather [33]	96 192 336 720	36696 36600 36456 36072	96	21

Table A.2: Multi-task forecasting datasets.

Category	Dataset	# classes	N	L	C
Finance	SharePriceIncrease [5]	2	965	60	1
Audio	JapaneseVowels [3] SpokenArabicDigits [3] Heartbeat [3]	9 10 2	270 6599 204	29 93 405	12 13 61
ECG	ECG5000 [5] NonInvasiveFetalECGThorax1 [5]	5 52	500 1800	140 750	1 1
EEG	Blink [3] FaceDetection [3] SelfRegulationSCP2 [3]	2 2 2	500 5890 200	510 62 1152	4 144 7
Sensors	ElectricDevices [5] Trace [5] FordB [5]	7 4 2	8926 100 3636	96 275 500	1 1 1
Human Activity	MotionSenseHAR [3] EMOPain [3] UWaveGestureLibrary [3]	6 3 8	966 968 120	200 180 315	12 30 3
Traffic	Chinatown [5] MelbournePedestrian [5] PEMS-SF [3]	2 10 7	20 1194 267	24 24 144	1 1 963

Table A.3: Multi-task classification datasets.

A.2.2 Few-shot Learning

For TS forecasting, classification, imputation, and anomaly detection in a few-shot setting, we evaluate the effectiveness of our proposed method using nine datasets for forecasting, six datasets for classification, four datasets for imputation, and five datasets for anomaly detection. The statistics of these datasets related to forecasting and classification are summarized in Table A.4, Table A.5, A.6, and A.7, respectively.

Category	Dataset	Prediction Length	N	L	C
Electricity	ETTh2 [39]	96	8449	96	7
		192	8353		
		336	8209		
		720	7825		
	ETTm1 [39]	96	34369	96	7
		192	34273		
		336	34129		
		720	33745		
Weather	SaugeenRiverFlow [21]	24	18921	48	1

Table A.4: Few-shot forecasting datasets.

Category	Dataset	# classes	N	L	C
ECG	ECG200 [5]	2	100	96	1
EEG	SelfRegulationSCP1 [3]	2	268	896	6
Human Activity	RacketSports [3]	4	151	30	6
	Handwriting [3]	26	150	152	3
	Epilepsy [3]	4	137	207	3
Sensor	StarLightCurves [5]	3	1000	1024	1

Table A.5: Few-shot classification datasets.

Category	Dataset	L	C
Electricity	ETTm1 [39]	96	7
	ETTh1 [39]	96	7
	ECL [33]	96	321
Weather	Weather [33]	96	21

Table A.6: Few-shot imputation datasets.

Category	Dataset	L	C
Machine	SMD [27]	96	38
	PSM [1]	96	25
Spacecraft	MSL [10]	96	55
	SMAP [10]	96	25
Infrastructure	SWaT [20]	96	51

Table A.7: Few-shot anomaly detection datasets.

A.2.3 Zero-shot Learning

For TS forecasting in a zero-shot setting, we evaluate the effectiveness of our proposed method using six datasets. Three of these datasets are used for the zero-shot setting with unseen datasets, while the remaining four datasets are used for the zero-shot setting with new prediction lengths. The statistics for the three unseen datasets are summarized in Table A.8.

Category	Dataset	Prediction Length	L	C
Electricity	Solar [24]	64	128	137
Weather	SaugeenRiverFlow [21]	128	256	1
Healthcare	Hospital [11]	16	32	767

Table A.8: Zero-shot forecasting datasets.

B Implementation Details

It is important to note that we follow the experimental settings of iTransformer for single-task and UniTS for multi-task settings, respectively. The following sections outline the specific settings we adhered to.

B.1 Implementation for Single-task Model: iTransformer

Following iTransformer [18], we use the Adam optimizer [13] and L2 loss for model optimization. The batch size is consistently set to 32, and the number of training epochs is fixed at 10. Since our approach is plug-and-play, we do not adjust any hyperparameters for our method; instead, we use the same hyperparameters employed by iTransformer.

B.2 Implementation for Multi-task Model: UniTS

Model architecture. In a multi-task setting, the UniTS network consists of three UniTS blocks, along with one `GEN` tower and one `CLS` tower. For each data source, specific prompt and task tokens are assigned, with forecasting tasks on the same source but with varying forecast lengths using the same prompt and `GEN` token. To enable zero-shot learning on new datasets, a shared prompt and `GEN` token are applied across all data sources. The embedding dimensions are set to 64 for the supervised version, and 32 for the prompt-tuning version, and all blocks in UniTS retain the same feature shape.

Model training. In multi-task settings, models are trained jointly on multiple tasks following a unified protocol. To match the largest dataset, samples from each dataset are repeated within each epoch. Supervised training is conducted over 5 epochs with gradient accumulation, yielding an effective batch size of 1024. The initial learning rate is set at 3.2e-2 and is adjusted using a multi-step decay schedule. For self-supervised pretraining, the models training with an are trained for 10 epochs with effective batch size of 4096, starting with a learning rate of 6.4e-3, which is adjusted using a cosine decay schedule.

C Related Works

C.1 MTS Forecasting Models

can be categorized into CI and CD models, where CI models process channels independently without accounting for dependencies between them, whereas CD models account for these dependencies. For CI models, DLinear [36] employs a linear model along the time dimension, and PatchTST [23] divides TS into patches and feeds them into a Transformer [30] in a CI manner, and PITS [16] combines channel independent and patch independent architectures with multi-layer perceptrons (MLPs). For CD models, Crossformer [37] employs a two-stage attention mechanism to capture both temporal and channel dependenciesand TSMixer [4] utilizes MLPs combined with patching to capture both dependencies. Recently, iTransformer [18] inverts the traditional Transformer framework in TS domain by treating each channel as a token instead of each patch, thereby shifting the focus from capturing temporal dependencies to channel dependencies. However, these models primarily focus on architectural solutions for handling CD and often overlook the characteristics of TS datasets, motivating us to consider CD varying across datasets.

C.2 TS Foundation Models

often borrow knowledge from other fields, such as natural language processing, primarily due to the lack of large-scale datasets in the TS domain. In response to this challenge, there have been efforts to adapt large language models (LLMs) for TS tasks: GPT4TS [40] fine-tunes the embedding layers of LLMs and Time-LLM [12] aligns TS data with LLM-based text prototypes to address TS tasks. Recent works have focused on pretraining TSFMs exclusively on TS datasets from various sources. MOMENT [9] and Timer [19] collect extensive and heterogeneous sets of TS datasets to pretrain Transformer-based TSFMs, while MOIRAI [31] enhances the Transformer architecture to address domain-specific challenges in constructing TSFMs. UniTS [7] proposes a TSFM that handles various tasks with a single architecture through prompt-tuning. However, these models do not account for the heterogeneity among datasets in terms of CD, while different TS datasets exhibit varying degrees of CD. This motivates us to adjust CD in TSFMs based on the characteristics of each dataset.

D Application to iTransformer

To demonstrate the effectiveness of our method, we apply our method to iTransformer [18] to solve TS forecasting tasks on 13 datasets. Table 3 presents the average MSE and MAE across four different horizons (H), showing consistent improvement across all datasets. Specifically, the performance gains in MSE on the PEMS datasets [17] (03, 04, 07, 08) are significantly large (12.7%, 19.0%, 19.6%, 40.2%), whereas the gains on the ETT datasets [39] (h1, h2, m1, m2) are relatively small (2.8%, 0.3%, 2.5%, 1.4%), suggesting a potential variation in the need for a CM across different datasets. Full results are described in Table D.1.

Metric		iTransformer		+ CM	
		MSE	MAE	MSE	MAE
ETTh1	96	0.387	0.405	**0.385**	**0.404**
	192	0.441	0.436	**0.438**	**0.434**
	336	0.491	0.462	**0.475**	**0.454**
	720	0.509	0.494	**0.477**	**0.474**
	Avg.	0.457	0.449	**0.444**	**0.441**
ETTh2	96	0.301	0.350	**0.295**	**0.347**
	192	0.381	0.399	**0.380**	**0.397**
	336	**0.423**	**0.432**	0.427	0.434
	720	**0.430**	0.446	0.432	**0.445**
	Avg.	0.384	0.407	**0.383**	**0.406**
ETTm1	96	0.342	0.377	**0.331**	**0.369**
	192	0.383	0.396	**0.372**	**0.390**
	336	0.418	0.418	**0.412**	**0.414**
	720	0.487	0.456	**0.479**	**0.453**
	Avg.	0.408	0.412	**0.398**	**0.406**
ETTm2	96	0.186	**0.272**	**0.184**	**0.272**
	192	0.254	0.314	**0.251**	**0.311**
	336	0.317	0.353	**0.312**	**0.350**
	720	0.416	0.409	**0.412**	**0.408**
	Avg.	0.293	0.337	**0.289**	**0.335**
Exchange	96	0.086	0.206	**0.085**	**0.205**
	192	0.181	0.303	**0.180**	**0.302**
	336	0.338	0.422	**0.337**	**0.421**
	720	0.869	0.704	**0.850**	**0.696**
	Avg.	0.368	0.409	**0.363**	**0.406**
Weather	96	0.174	0.215	**0.165**	**0.209**
	192	0.224	0.258	**0.213**	**0.251**
	336	0.281	0.298	**0.274**	**0.296**
	720	0.359	0.351	**0.350**	**0.346**
	Avg.	0.260	0.281	**0.250**	**0.275**
Solar	96	0.201	0.234	**0.197**	**0.231**
	192	0.238	0.263	**0.232**	**0.260**
	336	0.248	0.273	**0.241**	**0.270**
	720	0.249	0.275	**0.241**	**0.273**
	Avg.	0.234	0.261	**0.228**	**0.258**

Metric		iTransformer		+ CM	
		MSE	MAE	MSE	MAE
PEMS03	12	0.071	0.174	**0.063**	**0.168**
	24	0.097	0.208	**0.087**	**0.197**
	48	0.161	0.272	**0.133**	**0.250**
	96	0.240	0.338	**0.212**	**0.316**
	Avg.	0.142	0.248	**0.124**	**0.231**
PEMS04	12	0.081	0.188	**0.075**	**0.181**
	24	0.099	0.211	**0.086**	**0.196**
	48	0.133	0.246	**0.108**	**0.222**
	96	0.172	0.283	**0.125**	**0.242**
	Avg.	0.121	0.232	**0.098**	**0.210**
PEMS07	12	0.067	0.165	**0.061**	**0.157**
	24	0.088	0.190	**0.076**	**0.179**
	48	0.113	0.218	**0.086**	**0.188**
	96	0.140	0.246	**0.104**	**0.208**
	Avg.	0.102	0.205	**0.082**	**0.183**
PEMS08	12	0.088	0.193	**0.085**	**0.190**
	24	0.138	0.243	**0.126**	**0.234**
	48	0.334	0.353	**0.178**	**0.241**
	96	0.458	0.436	**0.221**	**0.260**
	Avg.	0.254	0.306	**0.152**	**0.231**
ECL	96	0.148	0.240	**0.140**	**0.235**
	192	0.167	0.258	**0.158**	**0.252**
	336	0.179	0.272	**0.172**	**0.267**
	720	0.220	0.310	**0.202**	**0.295**
	Avg.	0.179	0.270	**0.168**	**0.262**
Traffic	96	0.395	0.268	**0.391**	**0.266**
	192	0.417	0.277	**0.409**	**0.275**
	336	0.433	0.283	**0.426**	**0.282**
	720	0.467	0.300	**0.460**	**0.300**
	Avg.	0.428	0.282	**0.422**	**0.281**

Table D.1: TS forecasting results with 13 datasets.

E Application to UniTS

To demonstrate the effectiveness of our method on a TS foundation model, we apply it to four different TS tasks using UniTS [7] on datasets from various domains, under multiple settings, including multi-task, few-shot, and zero-shot settings. All experimental settings follow those outlined in UniTS [7]. The sections and tables outlining the full experiment results are listed in Table E.1.

Settings	Section	TS downstream tasks			
		FCST	CLS	IMP	AD
Multi-task	E.1	Table 2	Table E.2	-	-
Few-shot	E.3	Table E.4,E.5,E.6	Table E.7,E.8,E.9	Table E.10	Table E.11
Zero-shot	E.4	Table 2,2	-	-	-

Table E.1: Summary of experiments.

E.1 Multi-task Learning

For experiments under multi-task settings, we perform 20 TS forecasting and 18 classification tasks, where the full results are shown in Tables 2 and E.2, respectively.

18 Tasks	Shared (1 model)				Task-specific (18 models)					
	UniTS + CM		UniTS		iTransformer	TimesNet	PatchTST	Pyraformer	Autoformer	GPT4TS
	Sup.	PT	Sup.	PT	Sup.					FT
Heartbeat	67.3	70.2	59.0	69.3	66.8	**72.7**	65.9	**72.7**	71.7	69.8
JapaneseVowels	94.1	93.2	93.5	90.8	95.9	**97.6**	94.1	85.4	94.1	94.6
PEMS-SF	83.2	82.1	83.2	85.0	83.2	77.5	**83.8**	83.2	79.2	79.2
SelfRegulationSCP2	**58.3**	51.7	47.8	53.3	48.9	52.8	48.9	56.7	45.0	45.6
SpokenArabicDigits	97.1	93.5	97.5	92.0	97.8	**98.7**	97.5	92.1	97.3	97.5
UWaveGestureLibrary	**84.4**	83.8	79.1	75.6	82.2	**84.4**	81.9	72.2	42.2	81.9
ECG5000	93.4	93.4	92.6	93.4	93.3	92.6	**94.3**	91.4	91.9	93.0
NonInvasiveFetalECGThorax1	89.5	55.2	**90.5**	27.1	88.2	88.9	86.5	21.4	21.7	89.7
Blink	**99.1**	95.6	**99.1**	91.1	93.3	87.6	89.6	88.2	63.1	92.4
FaceDetection	64.7	54.6	64.1	57.6	66.0	66.2	63.9	**67.3**	59.2	66.1
ElectricDevices	62.4	60.5	60.3	55.4	57.3	49.5	59.5	**65.4**	56.1	62.9
Trace	**99.0**	93.0	91.0	82.0	79.0	91.0	77.0	74.0	60.0	96.0
FordB	**76.2**	64.2	76.0	62.8	72.7	68.9	61.4	55.3	66.4	77.7
MotionSenseHAR	92.8	**94.3**	92.8	93.2	93.6	90.6	75.8	88.7	30.2	96.2
EMOPain	75.5	80.8	78.0	80.3	79.4	78.0	79.2	**81.4**	69.9	79.4
Chinatown	97.7	**98.0**	97.7	**98.0**	97.4	97.7	97.7	27.4	96.8	96.5
MelbournePedestrian	89.3	78.3	87.3	77.0	89.3	**95.7**	80.4	52.3	75.0	94.0
SharePriceIncrease	62.9	66.6	61.9	**68.4**	61.9	65.0	68.0	63.1	61.5	63.7
1st Count (/18)	5	2	2	2	0	5	2	4	0	-
2nd Count (/18)	6	5	3	1	5	2	2	2	1	-
Average Score	**82.0**	78.3	80.6	75.1	80.3	80.9	78.1	68.8	65.6	82.0

Table E.2: Results of multi-task classification.

E.2 Few-shot Settings: Summary Results

For the tasks under the few-shot settings, we conduct four different tasks (FCST, CLS, IMP, AD), following the experimental settings of UniTS.

Few-shot FCST and CLS. We experiment nine forecasting tasks and six classification tasks under the few-shot settings with data ratios of 5%, 15%, and 20%. Table E.3a presents the results, which indicates that our method outperforms both iTransformer and UniTS in both PT and fine-tuning (FT) settings.

Few-shot IMP. We experiment six imputation tasks under the few-shot setting with a data ratio of 10%, where the goal is to impute 25% and 50% of missing data points. Table E.3b presents the results, indicating that our method outperforms UniTS and other state-of-the-art (SOTA) single-task models [32, 23, 18] in both PT and FT settings.

Few-shot AD. We experiment five anomaly detection tasks under the few-shot setting with a data ratio of 5%, where the results in Table E.3c indicate that our method outperforms UniTS and other SOTA methods in both PT and FT settings.

Ratio	Model		MSE	Acc.
5%	iTransformer	FT	0.598	51.4
	UniTS	PT	0.549	49.4
		FT	0.505	53.8
	UniTS + CM	PT	0.546	**54.9**
		FT	**0.489**	54.8
15%	iTransformer	FT	0.524	56.5
	UniTS	PT	0.525	53.2
		FT	0.487	59.7
	UniTS + CM	PT	0.522	55.4
		FT	**0.481**	**60.4**
20%	iTransformer	FT	0.510	59.9
	UniTS	PT	0.525	58.9
		FT	0.486	63.6
	UniTS + CM	PT	0.453	60.0
		FT	**0.425**	**64.8**

(a) 9 FCST and 6 CLS tasks.

Ratio	Model		MSE
25%	TimesNet	FT	0.246
	PatchTST	FT	0.191
	iTransformer	FT	0.186
	UniTS	PT	0.179
		FT	0.167
	UniTS + CM	PT	0.179
		FT	**0.158**
50%	TimesNet	FT	0.292
	PatchTST	FT	0.236
	iTransformer	FT	0.226
	UniTS	PT	0.232
		FT	0.213
	UniTS + CM	PT	0.225
		FT	**0.201**

(b) 6 IMP tasks.

Model		F_1
ADTrans.	-	79.2
TimesNet	FT	74.2
PatchTST	FT	84.3
iTransformer	FT	83.1
UniTS	PT	81.7
	FT	85.6
UniTS + CM	PT	82.0
	FT	**86.6**

(c) 5 AD tasks.

Table E.3: Four tasks under few-shot settings.

E.3 Few-shot Settings: Full Results

For the few-shot tasks, we conduct four distinct tasks: FCST, CLS, IMP, and AD, which are discussed in Sections E.3.1, E.3.2, E.3.3, and E.3.4, respectively.

E.3.1 Few-shot Forecasting

The results of few-shot forecasting with data ratios of 5%, 15%, and 20% are shown in Tables E.4, E.5, and E.6, respectively.

5%		iTransformer		UniTS				UniTS + CM			
		FT		PT		FT		PT		FT	
Data	H	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2	96	0.554	0.500	**0.405**	**0.417**	0.418	0.424	0.421	0.427	0.421	0.425
	192	0.440	0.438	0.400	0.406	0.377	0.397	0.386	0.402	**0.370**	**0.389**
	336	0.478	0.467	0.425	0.433	0.420	0.433	0.423	0.431	**0.416**	**0.425**
	720	0.483	0.480	0.446	0.457	0.439	0.452	**0.424**	0.444	0.428	**0.443**
RiverFlow	24	1.141	0.514	1.115	0.504	1.112	0.504	**1.097**	0.503	**1.097**	**0.500**
ETTm1	96	0.504	0.462	0.436	0.434	0.384	0.404	0.428	0.436	**0.354**	**0.384**
	192	0.555	0.485	0.462	0.448	0.414	0.418	0.475	0.458	**0.393**	**0.405**
	336	0.567	0.496	0.560	0.494	0.453	0.442	0.550	0.493	**0.420**	**0.423**
	720	0.659	0.539	0.703	0.558	0.526	0.483	0.689	0.554	**0.483**	**0.455**
Average		0.598	0.487	0.549	0.461	0.505	0.440	0.546	0.462	**0.489**	**0.429**

Table E.4: Results of few-shot forecasting (5%).

15%		iTransformer		UniTS				UniTS + CM			
		FT		PT		FT		PT		FT	
Data	H	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2	96	0.441	0.440	0.403	0.412	**0.399**	**0.409**	0.416	0.423	0.403	0.411
	192	0.398	0.410	0.396	0.404	0.394	0.399	0.388	0.403	**0.387**	**0.399**
	336	0.436	0.441	0.432	0.435	0.441	0.435	**0.419**	0.435	0.430	**0.431**
	720	0.438	0.453	0.448	0.457	0.449	0.453	**0.415**	**0.442**	0.433	0.446
RiverFlow	24	**1.067**	**0.467**	1.077	0.492	1.069	0.489	1.073	0.492	1.072	0.487
ETTm1	96	0.423	0.419	0.407	0.420	0.353	0.386	0.408	0.426	**0.342**	**0.380**
	192	0.464	0.439	0.434	0.432	0.384	0.400	0.449	0.447	**0.377**	**0.399**
	336	0.492	0.457	0.490	0.464	0.416	0.420	0.502	0.475	**0.406**	**0.148**
	720	0.558	0.493	0.641	0.537	0.480	0.455	0.621	0.530	**0.470**	**0.451**
Average		0.524	0.450	0.525	0.450	0.487	0.428	0.522	0.452	**0.481**	**0.425**

Table E.5: Results of few-shot forecasting (15%).

20%		iTransformer		UniTS				UniTS + CM			
		FT		PT		FT		PT		FT	
Data	H	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2	96	0.418	0.426	0.411	0.414	**0.391**	**0.405**	0.411	0.422	0.395	0.409
	192	0.395	0.407	0.383	**0.398**	0.395	0.403	**0.381**	0.400	0.390	0.400
	336	0.431	0.438	**0.419**	0.431	0.430	**0.430**	0.423	**0.430**	0.438	0.433
	720	0.431	0.449	0.440	0.453	0.444	0.449	**0.418**	**0.422**	0.456	0.456
RiverFlow	24	**1.056**	**0.462**	1.069	0.487	1.069	0.489	1.071	0.487	1.067	0.489
ETTm1	96	0.408	0.410	0.409	0.421	0.344	0.379	0.403	0.425	**0.339**	**0.376**
	192	0.444	0.428	0.443	0.439	0.377	0.397	0.450	0.450	**0.375**	**0.396**
	336	0.471	0.445	0.505	0.472	0.408	0.418	0.507	0.481	**0.403**	**0.415**
	720	0.536	0.482	0.648	0.536	0.472	0.453	0.621	0.531	**0.466**	**0.448**
Average		0.510	0.438	0.525	0.450	0.486	0.425	0.521	0.453	**0.482**	**0.425**

Table E.6: Results of few-shot forecasting (20%).

E.3.2 Few-shot Classification

The results of few-shot classification with data ratios of 5%, 15%, and 20% are shown in Tables E.7, E.8, and E.9, respectively.

5%	iTransformer	UniTS		UniTS + CM	
	FT	PT	FT	PT	FT
ECG200	78.0	67.0	77.0	**80.0**	77.0
Handwriting	5.4	4.6	4.7	4.8	**5.5**
SelfRegulationSCP1	62.8	66.2	74.7	**77.8**	73.7
RacketSports	37.5	31.6	35.5	39.5	**47.4**
Epilepsy	39.9	44.9	47.1	44.9	**57.2**
StarLightCurves	85.1	82.3	83.8	**86.3**	85.4
Average	51.4	49.4	53.8	**54.9**	54.8

Table E.7: Results of few-shot classification (5%).

15%	iTransformer	UniTS		UniTS + CM	
	FT	PT	FT	PT	FT
ECG200	81.0	74.0	78.0	73.2	**82.0**
Handwriting	**9.8**	7.3	8.1	9.2	8.5
SelfRegulationSCP1	67.9	59.0	**76.5**	69.3	68.6
RacketSports	**54.6**	40.1	50.7	44.7	51.3
Epilepsy	41.3	52.9	58.0	61.6	**68.1**
StarLightCurves	84.2	85.8	**87.1**	85.9	85.5
Average	56.5	53.2	59.7	55.4	**60.4**

Table E.8: Results of few-shot classification (15%).

20%	iTransformer	UniTS		UniTS + CM	
	FT	PT	FT	PT	FT
ECG200	81.0	76.0	77.0	**85.0**	82.0
Handwriting	**11.8**	8.0	8.5	7.6	9.8
SelfRegulationSCP1	77.1	68.6	70.6	**77.8**	74.4
RacketSports	54.6	51.3	**57.9**	38.8	50.7
Epilepsy	62.3	81.9	72.5	**84.1**	61.6
StarLightCurves	84.8	87.3	86.0	**90.0**	87.8
Average	59.9	58.9	63.6	60.0	**64.8**

Table E.9: Results of few-shot classification (20%).

E.3.3 Few-shot Imputation

The results of few-shot imputation with data ratios of 25% and 50% are shown in Table E.10

Ratio			ECL	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Avg.
25%	TimesNet	FT	0.245	0.369	0.193	0.442	0.119	0.106	0.246
	PatchTST	FT	0.195	0.315	0.147	0.309	0.092	0.089	0.191
	iTransformer	FT	0.174	0.301	0.185	0.254	0.113	0.087	0.186
	UniTS	PT	0.139	0.311	0.178	0.268	0.102	0.078	0.179
		FT	0.160	0.284	0.150	0.241	**0.090**	0.077	0.167
	UniTS + CM	PT	0.139	0.310	0.176	0.262	0.100	0.078	0.179
		FT	**0.129**	**0.275**	**0.149**	**0.231**	**0.090**	**0.073**	**0.158**
50%	TimesNet	FT	0.258	0.412	0.211	0.607	0.140	0.125	0.292
	PatchTST	FT	0.230	0.353	0.175	0.442	**0.111**	0.105	0.236
	iTransformer	FT	0.203	0.332	0.205	0.372	0.136	0.106	0.226
	UniTS	PT	0.172	0.352	0.251	0.380	0.134	0.103	0.232
		FT	0.191	0.322	0.198	0.352	0.118	0.095	0.213
	UniTS + CM	PT	0.162	0.353	0.240	0.370	0.128	0.097	0.225
		FT	**0.151**	**0.307**	**0.197**	**0.345**	0.116	**0.093**	**0.201**

Table E.10: Results of few-shot imputation.

E.3.4 Few-shot Anomaly Detection

The results of few-shot anomaly detection with data ratio of 5% are shown in Table E.11.

		MSL	PSM	SMAP	SMD	SWAT	Avg.
Anomaly Trans.	-	78.0	90.2	68.3	77.8	81.5	79.2
TimesNet	FT	33.9	91.0	68.5	84.0	**93.4**	74.2
iTransfomer	FT	80.4	96.5	67.2	82.4	89.0	83.1
PatchTST	FT	79.9	96.6	68.7	83.8	92.6	84.3
UniTS	PT	73.2	95.5	65.9	81.2	92.9	81.7
	FT	**81.3**	**97.3**	71.6	85.5	92.5	85.6
UniTS + CM	PT	73.7	95.5	66.0	82.0	92.9	82.0
	FT	**81.3**	**97.3**	**75.9**	**86.2**	92.6	**86.6**

Table E.11: Results of few-shot anomaly detection.

E.4 Zero-shot Learning

We perform TS forecasting tasks under two types of zero-shot settings: 1) *Zero-shot dataset*: We evaluate our model on an unseen dataset that was not included during training. 2) *Zero-shot task*: We assess the model's ability to predict a new forecasting horizon that was not encountered during training, by adding the mask tokens at the end of the TS to predict the desired future time steps.

Zero-shot dataset. For the TS forecasting task on unseen datasets, we evaluate our method using three datasets [24, 21, 11]. Table E.12a presents the results, demonstrating a performance gain when using the CM compared to not using it.

Zero-shot horizon. For the TS forecasting task with new forecasting horizons, we predict additional 384 time steps (by adding 24 masked tokens of length 16 at the end of the TS) on top of the base forecasting horizon of 96. Table E.12b shows the results with four different datasets [39, 33], showing performance gain on three out of four datasets.

Dataset	UniTS		+ CM		Impr.	
	MSE	MAE	MSE	MAE	MSE	MAE
Solar	0.597	0.607	**0.586**	**0.585**	**1.9%**	**3.6%**
River	**1.374**	0.698	**1.374**	**0.686**	0.0%	**1.7%**
Hospital	1.067	0.797	**1.020**	**0.777**	**4.4%**	**2.5%**
Avg.	1.013	0.701	**0.993**	**0.683**	**2.0%**	**2.6%**

(a) Zero-shot dataset.

Dataset	UniTS		+ CM		Impr.	
	MSE	MAE	MSE	MAE	MSE	MAE
ECL	0.237	0.329	**0.231**	**0.323**	**2.5%**	**1.8%**
ETTh1	0.495	0.463	**0.492**	0.463	**0.6%**	0.0%
Traffic	0.632	0.372	**0.592**	**0.369**	**6.3%**	**0.8%**
Weather	0.335	0.336	0.335	0.336	0.0%	0.0%

(b) Zero-shot horizon.

Table E.12: Zero-shot FCST tasks.

F Application to TimeSiam

To demonstrate the effectiveness of our proposed model on TimeSiam [6], which uses a self-supervised pretraining framework for TS with Siamese networks, we conduct experiments with two datasets that vary in channel size: Exchange, with a small number of channels (8), and ECL, with a large number of channels (321). Specifically, we apply variants of our method by using the domain parameter only during the fine-tuning stage and during both pretraining and fine-tuning stages. The results, shown in Table F.1, validate both components of our method, with the best performance achieved when using domain parameters at both pretraining and fine-tuning stages.

		TimeSiam		+ CM					
Correlation matrix		-		✓		✓		✓	
Domain parameters	Pretrain	-		-		-		✓	
	Fine-tune	-		-		✓		✓	
Dataset	H	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Exchange ($C=8$)	96	0.092	0.215	<u>0.089</u>	**0.207**	**0.088**	**0.207**	**0.088**	<u>0.209</u>
	192	**0.182**	0.306	**0.182**	<u>0.304</u>	**0.182**	**0.303**	**0.182**	0.305
	336	0.341	0.426	0.336	<u>0.422</u>	<u>0.332</u>	**0.417**	**0.329**	**0.417**
	720	0.806	0.679	0.792	0.670	<u>0.788</u>	<u>0.668</u>	**0.783**	**0.666**
	Avg.	0.356	0.407	0.350	0.401	<u>0.349</u>	<u>0.399</u>	**0.346**	**0.398**
ECL ($C=321$)	96	0.147	0.239	**0.140**	**0.236**	**0.140**	**0.236**	<u>0.141</u>	<u>0.237</u>
	192	0.162	0.253	**0.157**	<u>0.251</u>	**0.157**	<u>0.251</u>	**0.157**	**0.250**
	336	0.175	0.269	<u>0.173</u>	<u>0.268</u>	<u>0.173</u>	<u>0.268</u>	**0.172**	**0.267**
	720	0.215	0.304	**0.203**	<u>0.297</u>	**0.203**	<u>0.297</u>	**0.203**	**0.296**
	Avg.	0.175	0.266	**0.168**	<u>0.263</u>	**0.168**	<u>0.263</u>	**0.168**	**0.262**

Table F.1: Results of TS forecasting with TimeSiam.

G Further Analysis

CD ratio comparison. Table G.1 presents the CD ratios of CMs with and without[2] domain parameters ($r(\mathbf{M})$ and $r(|\mathbf{R}|)$), when using UniTS. The results show that while datasets with higher $r(|\mathbf{R}|)$ generally have higher $r(\mathbf{M})$, this relationship is not consistent; for instance, Weather [33] exhibits lower CD despite having a stronger correlation compared to ETTh1 [39]. Figure G.1 supports these findings by visualizing the channels of the datasets, revealing that the channels of ETTh1 tend to be more dependent on each other than those of Weather. These results underscore the importance of using domain parameters to adjust $|\mathbf{R}|$ for learning absolute dependencies specific to each dataset. Furthermore, datasets with a larger number of channels (C) tend to have higher $r(\mathbf{M})$, which aligns with the prior work [2] emphasizing CD over CI for datasets with more channels.

Domain params.		✗	✓
Dataset	C	$r(\|\mathbf{R}\|)$	$r(\mathbf{M})$
Weather	21	0.296 (2)	0.587 (1)
ILI	21	0.708 (7)	0.706 (2)
ETTh1	7	0.222 (1)	0.717 (3)
Exchange	21	0.513 (4)	0.749 (4)
ECL	321	0.489 (3)	0.800 (5)
Traffic	862	0.564 (5)	0.808 (6)
NN5	111	0.584 (6)	0.857 (7)

Table G.1: CD ratio comparison with rank.

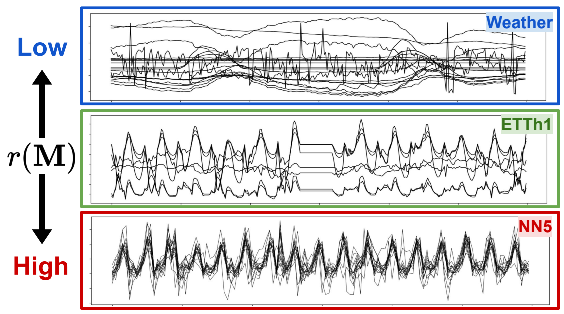

Figure G.1: TS visualization by $r(\mathbf{M})$.

Effectiveness of domain parameters. To demonstrate the importance of domain parameters in reflecting the degree of CD, we compare the CD ratio and the performance gain achieved with the CD framework against the CI framework with UniTS. Figure G.2 shows that the gain is highly correlated with the CD ratio of a CM with the domain parameters ($r(\mathbf{M})$), but less so without them ($r(|\mathbf{R}|)$).

Domain parameters for unseen dataset. For an unseen dataset, selecting the appropriate domain parameters is challenging, as these parameters are not learned during training. To address this issue, we propose three strategies: 1) averaging the parameters across all datasets, 2) averaging the parameters from the forecasting datasets, and 3) selecting parameters from the dataset with the closest $r(\bar{\mathbf{R}})$. Table G.2 demonstrates the robustness of these strategies, consistently outperforming UniTS.

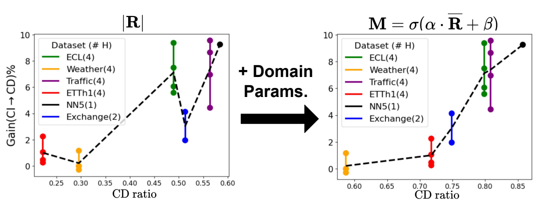

Figure G.2: Performance gain by CD vs. CD ratio.

Method	Dataset	MSE	MAE
UniTS		1.006	0.701
UniTS + CM	FCST + CLS	0.995	0.684
	FCST	**0.993**	**0.683**
	Closest	**0.993**	**0.683**

Table G.2: Domain params for unseen datasets.

[2]For a CM without domain parameters, we use the absolute correlation matrix ($|\mathbf{R}|$) instead of its zero-centered scaled version ($\bar{\mathbf{R}}$) to ensure a fair comparison with $\mathbf{M}$, which is also scaled between 0 and 1.

Visualization of CM. Figure G.3 shows the CMs of ECL [33] and ETTh1 [39], illustrating the dependencies between the channels of each dataset. The CM of ETTh1 reveals a hidden relationship between the first and fifth channels when using domain parameters, which is not identified by the correlation matrix alone.

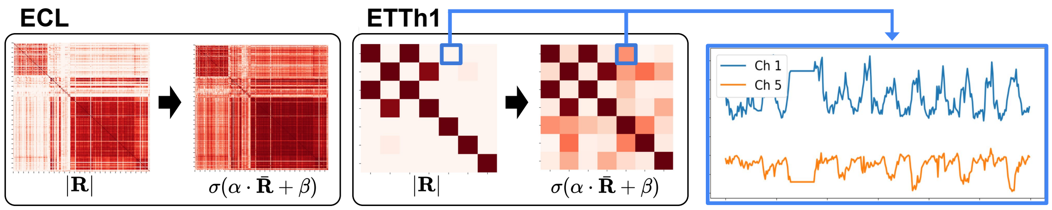

Figure G.3: **Visualization of CMs w/ and w/o domain parameters**. The figure shows the correlation matrices and the CMs of two datasets, with each color scaled from 0 (light) to 1 (dark).

Various TS metrics. To demonstrate the effectiveness of CMs using metrics beyond correlation, we apply CMs to iTransformer with three different metrics: 1) Euclidean distance (Euc.), which we min-max normalize to the range (0,1) and subtract from 1 to convert it into a similarity metric; 2) cosine similarity (Cos.), for which we take the absolute value, following the same intuition as correlation; and 3) dynamic time warping (DTW), where we apply the same process as with the Euclidean distance. Table G.3 presents the TS forecasting result in terms of average MSE for four different horizons, indicating that CMs yield a performance gain regardless of the metric used, with the best performance achieved with correlation. Note that we use DTW only for datasets with fewer than 100 channels due to its computational complexity.

Dataset	w/o CM	w/ CM			
		Euc.	Cos.	DTW	Corr.
ETTh1	0.457	0.445	0.446	0.444	**0.444**
ETTh2	0.384	0.384	0.384	0.385	**0.383**
ETTm1	0.408	0.402	0.403	0.401	**0.398**
ETTm2	0.293	0.292	0.290	0.292	**0.289**
PEMS03	0.142	0.146	0.134	-	**0.124**
PEMS04	0.121	0.111	0.105	-	**0.098**
PEMS07	0.102	0.092	0.087	-	**0.082**
PEMS08	0.254	0.163	0.179	-	**0.152**
Exchange	0.368	0.364	0.363	0.364	**0.363**
Weather	0.260	0.256	0.255	0.254	**0.250**
Solar	0.234	0.232	0.229	-	**0.228**
ECL	0.179	0.173	0.171	-	**0.168**
Traffic	0.428	0.432	0.443	-	**0.422**
Avg.	0.279	0.269	0.268	-	**0.261**

Table G.3: Various metrics for CMs.

Masked channel prediction. To evaluate the model's ability to capture CD, we introduce a novel evaluation method, *masked channel prediction*, which involves predicting the future values of the masked channel using the historical values of the unmasked channels. Specifically, we calculate the average loss for each channel when masked once, with the loss for the c-th channel expressed as:

$$L_{(c)}(y, \hat{y}) = \text{MSE}(y[:, c], \hat{y}_{(c)}[:, c]), \quad \text{where } \hat{y}_{(c)} = f(x_{(c)}), \tag{G.1}$$

where $x_{(c)}$ is x with the c-th channel masked, and $\hat{y}_{(c)}$ is the predicted output using $x_{(c)}$ as the input. Note that masked channel prediction is an *evaluation method* that does not require additional training, and instead uses a model pretrained without any masking.

To assess the effectiveness of CMs in capturing CD, we experiment masked channel prediction with iTransformer with and without CMs, imputing the historical values of the masked channels with there average values, which are essentially zero with normalization.

Tables G.4 and G.5 show the results of masked channel prediction for five datasets [33, 17], indicating significant improvement when a CM is applied to iTransformer compared to when it is not used. Furthermore, Figure G.4 visualizes the predicted result of the first channel of PEMS08, showing that the model with the CM predicts more accurately than without the CM.

Horizon	Exchange			ECL		
	Avg. MSE(C1∼C8)			Avg. MSE(C1∼C321)		
	iTrans.	+ CM	Impr.	iTrans.	+ CM	Impr.
96	0.139	**0.138**	**1.2%**	0.846	**0.526**	**37.8%**
192	0.236	**0.232**	**1.5%**	0.849	**0.563**	**33.7%**
336	0.383	**0.374**	**2.4%**	0.861	**0.594**	**31.0%**
720	0.934	**0.917**	**1.8%**	0.891	**0.741**	**16.8%**
Avg.	0.423	**0.415**	**1.8%**	0.862	**0.606**	**29.7%**

Table G.4: Results of masked channel prediction (Exchange, ECL).

Horizon	PEMS04			PEMS07			PEMS08		
	Avg. MSE(C1∼C307)			Avg. MSE(C1∼C883)			Avg. MSE(C1∼C170)		
	iTrans.	+ CM	Impr.	iTrans.	+ CM	Impr.	iTrans.	+ CM	Impr.
12	0.549	**0.300**	**45.4%**	0.835	**0.343**	**58.9%**	0.628	**0.200**	**68.1%**
24	0.718	**0.351**	**51.1%**	0.865	**0.448**	**48.1%**	0.678	**0.241**	**64.5%**
48	0.750	**0.409**	**45.5%**	1.038	**0.511**	**50.8%**	1.197	**1.059**	**11.5%**
96	0.758	**0.513**	**32.3%**	1.040	**0.640**	**38.5%**	1.375	**1.217**	**11.5%**
Avg.	0.694	**0.393**	**43.3%**	0.945	**0.486**	**48.6%**	0.970	**0.679**	**29.9%**

Table G.5: Results of masked channel prediction (PEMS datasets).

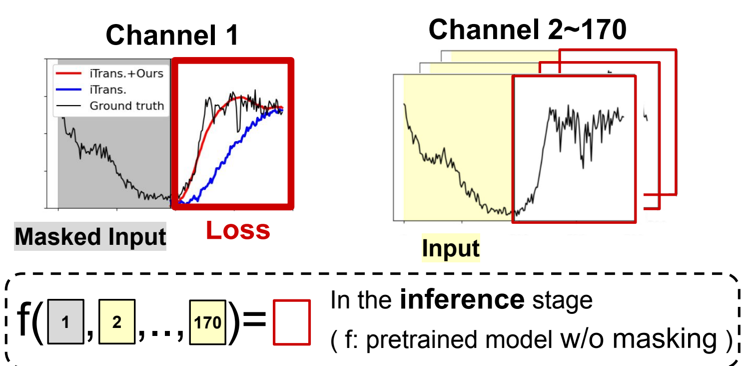

Figure G.4: Masked channel prediction.

Extending domain parameters. The proposed domain parameters α and β are scalars that adjust $\bar{\mathbf{R}}$ by changing its elements monotonically. For further flexibility, we design alternative options for the parameters: 1) a vector $\mathbf{E}$ for each channel and 2) a matrix $\mathbf{A}$ for each dataset. Both options are used to construct an adjustment matrix that is element-wise multiplied to $\bar{\mathbf{R}}$, as shown in Table G.6. The first option serves as identifiable vectors for each channel, with the adjustment matrix constructed based on the inner product between these vectors and normalized with $\mathrm{Norm}(\cdot) = \mathrm{Softmax}\left(\mathrm{ReLU}\left(\cdot\right)\right)$, while the second option acts as the adjustment matrix itself. For the vector parameters, we also implement an asymmetric matrix version that requires two different vectors for each channel: one for the inner vector ($\mathbf{E}_1$) and the other for the outer vector ($\mathbf{E}_2$), as described in the previous work [34]. Table G.7 shows that using scalar parameters achieves the best performance, demonstrating the efficiency of CMs by requiring only two additional parameters per dataset.

	Domain parameters	Channel mask ($\mathbf{M}$)	Asym.
Scalar	$\alpha, \beta \in \mathbb{R}^1$	$\sigma\left(\alpha \cdot \bar{\mathbf{R}} + \beta\right)$	✗
Vector	$\mathbf{E} \in \mathbb{R}^d$	$\mathrm{Norm}(\mathbf{E}\mathbf{E}^T) \odot \bar{\mathbf{R}}$	✗
	$\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^d$	$\mathrm{Norm}(\mathbf{E}_1\mathbf{E}_2^T) \odot \bar{\mathbf{R}}$	✓
Matrix	$\mathbf{A} \in \mathbb{R}^{C \times C}$	$\mathbf{A} \odot \bar{\mathbf{R}}$	✓

Table G.6: Extension of domain parameters.

	Average MSE across four horizons													Avg.
	ETTh1	ETTh2	ETTm1	ETTm2	PEMS03	PEMS04	PEMS07	PEMS08	Exchange	Weather	Solar	ECL	Traffic	
α, β	**0.444**	**0.383**	**0.398**	**0.289**	**0.124**	**0.098**	**0.082**	**0.152**	**0.363**	**0.250**	0.228	**0.168**	0.422	**0.261**
$\mathbf{E}$	0.452	0.391	0.402	0.291	0.150	0.106	0.096	0.202	0.364	0.255	0.234	0.177	0.416	0.272
$\mathbf{E_1}, \mathbf{E_2}$	0.452	0.391	0.402	0.291	0.152	0.105	0.095	0.205	0.364	0.255	0.233	0.177	**0.415**	0.272
$\mathbf{A}$	0.454	0.391	0.402	0.291	0.138	0.099	0.102	0.182	0.364	0.259	**0.226**	0.177	0.418	0.269
-	0.457	0.384	0.408	0.293	0.142	0.121	0.102	0.254	0.368	0.260	0.234	0.179	0.428	0.279

Table G.7: **Results of various domain parameters.** Using scalar domain parameters (α, β) which scale and shift the correlation matrix yields the best results.

Efficiency analysis. To demonstrate the efficiency of CMs, we compare the training and inference times of iTransformer on two datasets [33] with varying numbers of channels, using only attention matrices, only CMs, and both. Table G.8 indicates that incorporating CMs does not significantly impact computational time, even with datasets containing a large number of channels, with training time measured per epoch and inference time measured per data instance. It is important to note that correlation matrices can be precomputed offline, making CMs practical for use.

$L, H = 96$	Weather ($C = 21$)			ECL ($C = 321$)		
Channel mask	✓		✓	✓		✓
Attention matrix		✓	✓		✓	✓
Train (sec/epoch)	24.1	26.2	26.7	26.0	33.2	36.4
Inference (ms)	11.1	11.1	11.2	11.0	12.4	13.2
Avg. MSE	0.259	0.260	**0.250**	0.176	0.179	**0.168**

Table G.8: Efficiency analysis.

Robustness to missing values. To demonstrate the robustness of our method to missing values, we analyze scenarios where some TS values are randomly missing at ratios of 10%, 25%, 50%, and 75%, with the missing values linearly interpolated using adjacent values. Figure G.5 shows the result on ETTh2 [39] using iTransformer, indicating that both $r(|\mathbf{R}|)$ and the performance remain robust despite the missing values, making our method applicable in real-world scenarios.

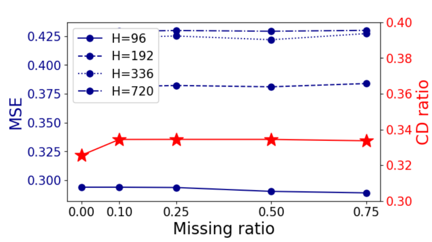

Figure G.5: Robustness to missingness.