Make Transformer Great Again for Time Series Forecasting: Channel Aligned Robust Dual Transformer

Wang Xue[*] **Tian Zhou**[*] **QingSong Wen** **Jinyang Gao** **Bolin Ding** **Rong Jin**
{xue.w,tian.zt,qingsong.wen,jinyang.gjy,bolin.ding,jinrong.jr}@alibaba-inc.com

Abstract

Recent studies have demonstrated the great power of deep learning methods, particularly Transformer and MLP, for time series forecasting. Despite its success in NLP and CV, many studies found that Transformer is less effective than MLP for time series forecasting. In this work, we design a special Transformer, i.e., channel-aligned robust dual Transformer (CARD for short), that addresses key shortcomings of Transformer in time series forecasting. First, CARD introduces a dual Transformer structure that allows it to capture both temporal correlations among signals and dynamical dependence among multiple variables over time. Second, we introduce a robust loss function for time series forecasting to alleviate the potential overfitting issue. This new loss function weights the importance of forecasting over a finite horizon based on prediction uncertainties. Our evaluation of multiple long-term and short-term forecasting datasets demonstrates that CARD significantly outperforms state-of-the-art time series forecasting methods, including both Transformer and MLP-based models.

1 Introduction

Multivariate time series forecasting has emerged as a crucial task in various domains such as weather prediction, financial investment, energy management, and traffic flow estimation. The rapid development of deep learning models has led to significant advancements in time series forecasting techniques, particularly in multivariate time series forecasting. Among various deep learning models developed for time series forecasting, both transformer and MLP-based models have demonstrated great performance thanks to their ability to capture complex long-term temporal dependencies (Zhou et al., 2021, 2022b; Wu et al., 2021; Liu et al., 2022a; Challu et al., 2022; Zeng et al., 2023; Wu et al., 2023; Zhang & Yan, 2023; Nie et al., 2023; Woo et al., 2022a,b; Liu et al., 2022b; Zhou et al., 2022a).

For multivariate time series forecasting, a model is expected to yield a better performance by exploiting the dependence among different prediction variables, so-called channel-dependent (CD) methods. However, multiple recent works (e.g., Nie et al. 2023; Zeng et al. 2023) show that in general channel-independent (CI) forecasting models (i.e., all the time series variables are forecast independently) outperform the CD models. Analysis from (Han et al., 2023) indicates that CI forecasting models are more robust while CD models have higher modeling capacity. Given that time series forecasting usually involves high noise levels, typical transformer-based forecasting models with CD design can suffer from the issue of overfitting noises, leading to limited performance.

Another type of popular forecasting model besides transformer is the MLP-based model. In 2019, (Oreshkin et al., 2020) proposed a bi-residual MLP model and beat the winning solution in M4

[*] Equal contribution

Preprint. Under review.

competition (Makridakis et al., 2018) with an ensemble strategy. Recently, (Zeng et al., 2023) argued that the numerical time series data lack semantics and transformers are ineffective in long-term forecasting. (Zeng et al., 2023) conducted various experiments on analyzing the efficiency of transformer structures in long-term forecasting tasks and showed that the simple linear model can outperform transformers when using longer inputs. These empirical studies and analyses raised an important question, i.e., whether the transformer is an effective structure for time series forecasting.

In this paper, we propose a Channeled Aligned Dual Transformer, or CARD for short, that effectively leverages the dependence among channels (i.e., forecasting variables) and alleviates the issue of overfitting noises in time series forecasting. Unlike typical transformers for time series analysis that only capture temporal dependency among signals through attention over tokens, the dual transformer model also takes attention across different variables and hidden dimensions, which captures the correlation among prediction variables and local information within each token. We observe that related approaches have been exploited in computer vision (Ding et al., 2022; Ali et al., 2021). To improve the robustness of the transformer for time series forecast, we further introduce an exponential smoothing layer over queries/keys tokens and a dynamic projection module when dealing with information among different channels. Finally, to alleviate the issue of overfitting noises, a robust loss function is introduced to weight each prediction by its uncertainty in the case of forecasting over a finite horizon. The overall model architecture is illustrated in Figure 1. We verify the effectiveness of the proposed model on various numerical benchmarks by comparing it to the state-of-the-art methods for transformer and MLP-based methods.

Here we summarized our key contributions as follows:

1. We propose a Channel Aligned Robust Dual Transformer (CARD) which efficiently and robustly aligns the information among different channels.
2. CARD demonstrates superior performance in seven benchmark datasets for long-term forecasting and the M4 dataset for short-term forecasting, outperforming the state-of-the-art models. Our studies have confirmed the effectiveness of the self-attention scheme.
3. We develop a robust signal decay-based loss function that utilizes signal decay to bolster the model's ability to concentrate on forecasting for the near future. Our empirical assessment has confirmed that this loss function is effective in improving the performance of other benchmark models as well.

The remainder of this paper is structured as follows. In Section 2, we provide a summary of related works relevant to our study. Section 3 presents the proposed detailed model architecture. Section 4 describes the loss function design with a theoretical explanation via maximum likelihood estimation of Gaussian and Laplacian distributions. In Section 5, we demonstrate the results of the numerical experiments in long-term/short-term time series forecasting benchmarks and conduct a comprehensive analysis to determine the effectiveness of the self-attention scheme for time series forecasting. Additionally, we discuss ablations and other experiments conducted in this study. Finally, in Section 6, the conclusions and future research directions are discussed.

2 Related Work

2.1 Transformers for Time Series Forecasting

There is a large body of work that tries to apply Transformer models to forecast long-term time series in recent years (Wen et al., 2023). We here summarize some of them. LogTrans (Li et al., 2019) uses convolutional self-attention layers with LogSparse design to capture local information and reduce space complexity. Informer (Zhou et al., 2021) proposes a ProbSparse self-attention with distilling techniques to extract the most important keys efficiently. Autoformer (Wu et al., 2021) borrows the ideas of decomposition and auto-correlation from traditional time series analysis methods. FEDformer (Zhou et al., 2022b) uses Fourier enhanced structure to get a linear complexity. Pyraformer (Liu et al., 2022a) applies pyramidal attention module with inter-scale and intra-scale connections which also get a linear complexity. LogTrans avoids a point-wise dot product between the key and query, but its value is still based on a single time step. Autoformer uses autocorrelation to get patch-level connections, but it is a handcrafted design that doesn't include all the semantic information within a patch. A recent work PatchTST (Nie et al., 2023) studies using a vision transformer type

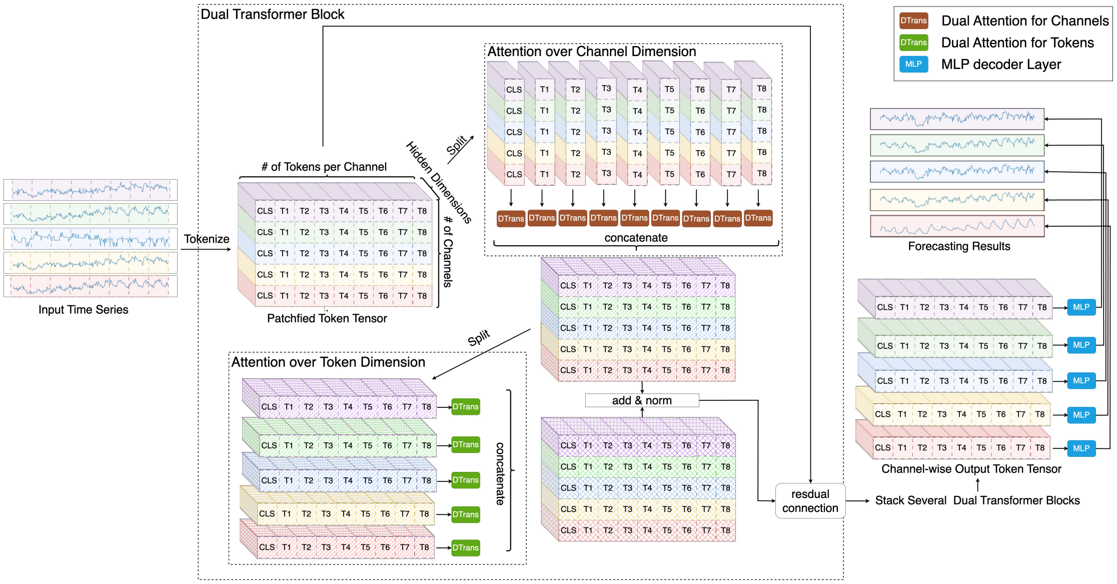

Figure 1: Illustration of the architecture of CARD.

model for long-term forecasting with channel independent design. The work closest to our proposed method is Crossformer (Zhang & Yan, 2023). This work utilizes a hierarchy attention mechanism to leverage cross-dimension dependencies and achieves moderate performance in the same benchmark datasets that we use in this work.

2.2 RNN, MLP and CNN Models for Time Series Forecasting

Besides transformers, other types of networks are also widely explored. For example, (Lai et al., 2018; Lim et al., 2021; Salinas et al., 2020; Smyl, 2020; Wen et al., 2017; Rangapuram et al., 2018; Zhou et al., 2022a; Gu et al., 2022) study the RNN/state-space models. In particular, (Smyl, 2020) considered equipping RNN with exponential smooth and first time beat the statistical models in forecasting tasks (Makridakis et al., 2018). (Chen et al., 2023; Oreshkin et al., 2020; Challu et al., 2022; Li et al., 2023; Zeng et al., 2023; Das et al., 2023; Zhang et al., 2022) explored MLP-type structures for time series forecasting. (Zeng et al., 2023) raises doubts about the effectiveness of a self-attention scheme for time series forecasting. However, due to the nature of the MLP layer, these models cannot effectively utilize the correlation/covariance information among subsequences which require bi-linear structures. CNN models (e.g., Wu et al. 2023; Wen et al. 2017; Sen et al. 2019) use the temporal convolution layer to extract the subsequence-level information. When dealing with multivariate forecasting tasks, the smoothness in adjacent covariates is assumed or the channel-independent strategy is used.

3 Model Architecture

The illustration of the architecture of CARD is shown in Figure 1. Let $\boldsymbol{a}_t \in \mathbb{R}^C$ be the observation of time series at time t with channel $C \geq 1$. Our objective is to use L recent historical data points (e.g., $\boldsymbol{a}_{t-L+1}, ..., \boldsymbol{a}_t$) to forecast the future T steps observations. (e.g., $\boldsymbol{a}_{t+1}, ..., \boldsymbol{a}_{t+T}$), where $L, T \geq 1$.

3.1 Tokenization

We adopt the idea of pacifying (Nie et al. 2023; Zhang & Yan 2023) to convert the input time series into token tensor. Let's denote $\boldsymbol{A} = [\boldsymbol{a}_{t-L+1}, ..., \boldsymbol{a}_t] \in \mathbb{R}^{C \times L}$ as the input data matrix, S and P as stride and patch length respectively. We unfold the matrix $\boldsymbol{A}$ into the raw token tensor $\tilde{X} \in \mathbb{R}^{C \times N \times P}$, where $N = \lfloor \frac{L-P}{S} + 1 \rfloor$. Here we convert the time series into several P length segments and each raw token maintains part of the sequence level semantic information which makes the attention scheme more efficient compared to the vanilla point-wise counterpart.

We then use a dense MLP layer $F_1 : P \to d$, a extra token $\mathbf{cls} \in \mathbb{R}^{C\times d}$ and positional embedding $\boldsymbol{E} \in \mathbb{R}^{C\times N\times d}$ to generate the token matrix as follows:

$$\boldsymbol{X} = [\mathbf{cls}, F_1(\tilde{\boldsymbol{X}}) + \boldsymbol{E}], \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{C\times(N+1)\times d}$ and d is the hidden dimension. Compared to Nie et al. (2023) and Zhang & Yan (2023), our token construction introduces a extra cls token. The cls token is an analogy to the *static covariate encoder* in Lim et al. (2021) and allows us to have a place to inject some statistic features.

3.2 Dual Attention over Tokens

We consider generating $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ via linear projection of the token tensor $\boldsymbol{X}$:

$$\boldsymbol{Q} = F_q(\boldsymbol{X}),\ \boldsymbol{K} = F_k(\boldsymbol{X}),\ \boldsymbol{V} = F_v(\boldsymbol{X}), \tag{2}$$

where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{C\times(N+1)\times d}$ and F_q, F_k, F_v are MLP layers.

We next convert $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ into $\{\boldsymbol{Q}_i\}$,$\{\boldsymbol{K}_i\}$,$\{\boldsymbol{V}_i\}$ where $\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i \in \mathbb{R}^{C\times(N+1)\times d_{\text{head}}}$, $i = 1, 2, ..., H$. H and d_{head} are number of heads and head dimension, respectively.

Besides the standard attention in tokens, we also introduce an extra attention structure in hidden dimensions that helps capture the local information within each patch. The attentions in both tokens and hidden dimensions are computed as follows:

$$\boldsymbol{A}_{i1} = \text{softmax}\left(\frac{1}{\sqrt{d}} \cdot \underset{cn_1k,cn_2k\to cn_1n_2}{\text{einsum}}(\text{EMA}(\boldsymbol{Q}_i), \text{EMA}(\boldsymbol{K}_i))\right) \tag{3}$$

$$\boldsymbol{A}_{i2} = \text{softmax}\left(\frac{1}{\sqrt{N}} \cdot \underset{cnk_1,cnk_2\to ck_1k_2}{\text{einsum}}(\boldsymbol{Q}_i, \boldsymbol{K}_i)\right), \tag{4}$$

where $\boldsymbol{A}_{i1} \in \mathbb{R}^{C\times(N+1)\times(N+1)}$, $A_{i2} \in \mathbb{R}^{C\times(N+1)\times C}$ and we use einsum and EMA to denote the Einstein Summation Convention and Exponential Moving Average (EMA), respectively.

By applying EMA on $\boldsymbol{Q}_i$ and $\boldsymbol{K}_i$, each query token will be able to gain higher attention scores on more key tokens and thus the output becomes more robust. Similar techniques are also explored in Ma et al. (2023) and Woo et al. (2022b). Different from those in the literature, we find that using a fixed EMA parameter that remains the same for all dimensions is enough to stabilize the training process. Thus, our EMA doesn't contain learnable parameters.

The output is computed as:

$$\boldsymbol{O}_{i1} = \underset{cnn_2,cn_2d\to cnd}{\text{einsum}}(\boldsymbol{A}_{i1}, \boldsymbol{V}_i), \quad \boldsymbol{O}_{i2} = \underset{cdk_2,cnk_2\to cnd}{\text{einsum}}(\boldsymbol{A}_{i2}, \boldsymbol{V}_i). \tag{5}$$

We next apply the batch normalization (Ioffe & Szegedy, 2015) to $\boldsymbol{O}_{i1}$ and $\boldsymbol{O}_{i2}$ to reshape the output scale. Finally, the residual connection structure is used to generate the final output of the dual attention block.

The total number of tokens is on the order of $\mathcal{O}(L/S)$ per channel and the complexity in attentions becomes $\mathcal{O}(C \cdot d^2 \cdot L^2/S^2)$, which is smaller than $\mathcal{O}(C \cdot d^2 \cdot L^2)$ complexity of the vanilla point-wise token construction. In practice, one can use efficient attention implementation (e.g., FlashAttention Dao et al. 2022) to further obtain nearly linear computational performance.

3.3 Dual Attention over Channels

We first compute $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ via (2). Due to the potential high-dimensionality issue of covariates, the vanilla method may suffer from computation overhead and overfitting. Take traffic dataset (PeMS) as an example, this dataset contains 862 covariates. When setting the lookback window size as 96, the

The tokens sequence in our setting is represented by tensor instead of matrix. The conventional matrix multiplication notation may cause confusion and we use Einstein Summation Convention instead.

Formally, an EMA operator recursively calculates the output sequence $\{\boldsymbol{y}_i\}$ w.r.t. input sequence $\{\boldsymbol{x}_i\}$ as $\boldsymbol{y}_t = \alpha\boldsymbol{x}_t + (1-\alpha)\boldsymbol{y}_{t-1}$, where $\alpha \in (0, 1)$ is the EMA parameter representing the degree of weighting decrease.

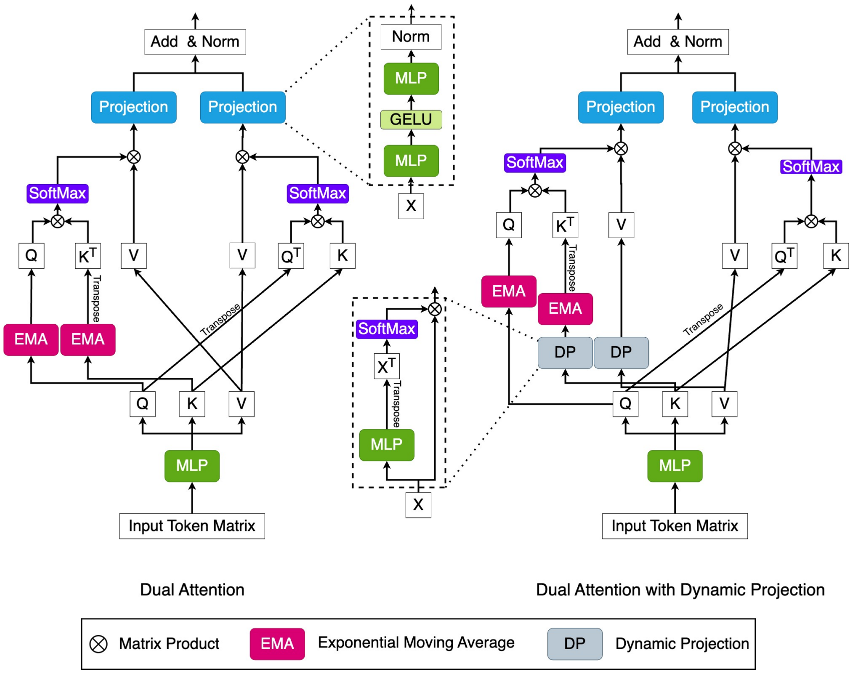

Figure 2: Architecture for the dual attention block in CARD.

attention over channels will require at least 80 times the computational cost of attention over tokens. The full attention will also merge a lot of noise patterns into the output token and lead to spurious correlation in the final forecasting results. In this paper, we consider using the dynamic projection technique (Zhu et al., 2021) to get "*summarized*" tokens to the $\boldsymbol{K}$ and $\boldsymbol{V}$ as shown in Figure 2. We first use MLP layers F_{pk} and F_{pv} to project hidden dimensions from d to some fixed r with $r \ll C$, and then we use $\mathrm{softmax}$ to normalized the projected tensors $\boldsymbol{P}_k$ and $\boldsymbol{P}_v$ as follow:

$$\boldsymbol{P}_k = \mathrm{softmax}(F_{pk}(\boldsymbol{K})), \quad \boldsymbol{P}_v = \mathrm{softmax}(F_{pv}(\boldsymbol{V})), \tag{6}$$

where $\boldsymbol{P}_k, \boldsymbol{P}_v \in \mathbb{R}^{C\times(N+1)\times r}$. Next the "*summarized*" tokens are computed by

$$\tilde{\boldsymbol{K}} = \underset{cnd,cnr\to rnd}{\mathrm{einsum}}(\boldsymbol{P}_k, \boldsymbol{K}), \quad \tilde{\boldsymbol{V}} = \underset{cnd,cnr\to rnd}{\mathrm{einsum}}(\boldsymbol{P}_V, \boldsymbol{V}). \tag{7}$$

Finally, the dual transformer over covariates is conducted by applying $\boldsymbol{Q}$, $\tilde{\boldsymbol{K}}$ and $\tilde{\boldsymbol{V}}$ to (3)-(5). The total computational cost is reduced to $\mathcal{O}(L/S \cdot C \cdot r \cdot d^2)$ which is smaller than $\mathcal{O}(L/S \cdot C^2 \cdot d^2)$ cost of the standard attention.

4 Signal Decay-based Loss Function

In this section, we discuss our loss function design. In literature, the Mean Squared Error (MSE) loss is commonly used to measure the discrepancy between the forecasting results and the ground truth observations. Let $\hat{\boldsymbol{a}}_{t+1}(\boldsymbol{A}), ..., \hat{\boldsymbol{a}}_{t+L}(\boldsymbol{A})$ and $\boldsymbol{a}_{t+1}(\boldsymbol{A}), ..., \boldsymbol{a}_{t+L}(\boldsymbol{A})$ be the predictions and real obversations from time $t+1$ to $t+L$ given historical information $\boldsymbol{A}$. The overall objective loss becomes:

$$\min \quad \mathbb{E}_{\boldsymbol{A}}\left[\frac{1}{L}\sum_{l=1}^{L}\|\hat{\boldsymbol{a}}_{t+l}(\boldsymbol{A}) - \boldsymbol{a}_{t+l}(\boldsymbol{A})\|_2^2\right]. \tag{8}$$

One drawback of plain MSE loss for forecasting tasks is that the different time steps' errors are equally weighted. In real practice, the correlation of historical information to far-future observations is usually smaller than that to near-future observations, implying that far-future observations have higher variance. Therefore, the near-future loss would contribute more to generalization improvement than the far-future loss. To see this, we assume that our time series follows the first-order Markov process, i.e., $\boldsymbol{a}_{t+1} \sim \mathcal{N}(G(\boldsymbol{a}_t), \sigma^2 I)$, where G is the smooth transition function with Lipschitz constant 1, $\sigma > 0$ and $t = 1, 2, ...$ Then, we have

$$\mathrm{var}(\boldsymbol{a}_{t+1}) = \mathrm{var}(G(\boldsymbol{a}_t)) + \sigma^2 I \preceq \mathrm{var}(\boldsymbol{a}_t) + \sigma^2 I, \tag{9}$$

where $\mathrm{var}(\boldsymbol{a})$ denote the covariance matrix of $\boldsymbol{a}$. By recursively using (9) from $t + L$ to t and for all $l \in [t, t + L]$, we have

$$\mathrm{var}(\boldsymbol{a}_{t+l}) \preceq l\sigma^2 I + \mathrm{var}(\boldsymbol{a}_t). \tag{10}$$

When $\boldsymbol{a}_t$ is already observed, we have $\mathrm{var}(\boldsymbol{a}_t) = 0$ and (10) implies $\mathrm{var}(\boldsymbol{a}_{t+l}) \preceq l\sigma^2 I.$ If we use negative log-likelihood estimation over Gaussian distribution, we come up with the following approximated loss function:

$$\begin{aligned} \min \quad & \mathbb{E}_{\boldsymbol{A}} \left[\frac{1}{2} \sum_{l=1}^{L} \left(\hat{\boldsymbol{a}}_{t+l}(\boldsymbol{A}) - \boldsymbol{a}_{t+l}(\boldsymbol{A}) \right)^{\top} \mathrm{var}\left(\boldsymbol{a}_{t+l} \right)^{-1} \left(\hat{\boldsymbol{a}}_{t+l}(\boldsymbol{A}) - \boldsymbol{a}_{t+l}(\boldsymbol{A}) \right) \right] \\ \geq & \mathbb{E}_{\boldsymbol{A}} \left[\frac{1}{2} \sum_{l=1}^{L} \frac{\| \hat{\boldsymbol{a}}_{t+l}(\boldsymbol{A}) - \boldsymbol{a}_{t+l}(\boldsymbol{A}) \|_2^2}{l\sigma^2} \right] \propto \mathbb{E}_{\boldsymbol{A}} \left[\frac{1}{L} \sum_{l=1}^{L} l^{-1} \| \hat{\boldsymbol{a}}_{t+l}(\boldsymbol{A}) - \boldsymbol{a}_{t+l}(\boldsymbol{A}) \|_2^2 \right]. \end{aligned} \tag{11}$$

Compared (11) to (8), the far-future loss is scaled down to address the high variance. Since Mean Absolute Error (MAE) is more resilient to outliers than square error, we propose to use the loss function in the following form:

$$\min \mathbb{E}_{\boldsymbol{A}} \left[\frac{1}{L} \sum_{l=1}^{L} l^{-1/2} \left\| \hat{\boldsymbol{a}}_{t+l}(\boldsymbol{A}) - \boldsymbol{a}_{t+l}(\boldsymbol{A}) \right\|_1 \right], \tag{12}$$

where (12) can be derived via (11) with replacing the Gaussian distribution by Laplace distribution.

5 Experiments

5.1 Long Term Forecasting

Datasets We conducted experiments on seven real-world benchmarks, including four Electricity Transform Temperature (ETT) datasets (Zhou et al., 2021) comprising of two hourly and two 15-minute datasets, one 10-minute weather forecasting dataset (Wetterstation), one hourly electricity consumption dataset (UCI), and one hourly traffic road occupancy rate dataset (PeMS).

Baselines and Experimental Settings We use the following recent popular models as baselines: Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022b), Nonstationary Transformer (Liu et al., 2022b), ETSformer (Woo et al., 2022b), FilM (Zhou et al., 2022a), LightTS (Zhang et al., 2022), MICN (Wang et al., 2023b), TimesNet (Wu et al., 2023), Dlinear (Zeng et al., 2023), Crossformer (Zhang & Yan, 2023), and PatchTST (Nie et al., 2023). We use the experimental settings in (Zhou et al., 2021; Wu et al., 2023) and keep the lookback length as 96 for fair comparisons. MSE and MAE results are reported. More details on model configurations and model code can be found in Appendix C and Appendix D, respectively.

Results The results are summarized in Table 1. Regarding the average performance across four different output horizons, CARD gains the best performance in 6 out of 7 and 7 out of 7 in MSE and MAE, respectively. In single-length experiments, CARD achieves the best results in **82%** cases in MSE metric and **100%** cases in MAE metric.

For problems with complex covariate structures, the proposed CARD method beats the benchmarks by significant margins. For instance, in Electricity (321 covariates), CARD consistently outperforms the second-best algorithm by reducing MSE/MAE by more than **9.0%** on average in each forecasting horizon experiment. By leveraging 21 covariates for Weather and 862 covariates for Traffic, we achieve a large reduction in MSE/MAE of over **7.5%**. This highlights CARD's exceptional capability to incorporate extensive covariate information for improved prediction outcomes. Furthermore, Crossformer (Zhang & Yan, 2023) employs a comparable concept of integrating cross-channel data to enhance predictive accuracy. Remarkably, CARD significantly reduces the MSE/MAE by over **20%** on 6 benchmark datasets compared to Crossformer, which shows our dual attention design is much more effective in utilizing cross-channel information.

It's important to note that while Dlinear shows strong performance in those tasks using an MLP-based model, CARD still consistently reduces MSE/MAE by **5%** to **27.5%** across all benchmark datasets.

Recent works, such as (Zeng et al., 2023; Nie et al., 2023), have shown that increasing the lookback length can improve performance. In our study, we also report the numerical performance of CARD with a longer lookback length in Appendix E, and CARD consistently outperforms all baseline models when prolonging input sequence as well, demonstrating significantly lower MSE errors across all benchmark datasets.

Table 1: Long-term forecasting tasks. The lookback length is set as 96. All models are evaluated on 4 different prediction horizons $\{96, 192, 336, 720\}$. The best model is in boldface and the second best is underlined.

Models		CARD		PatchTST		MICN		TimesNet		Crossformer		Dlinear		LightTS		FilM		ETSformer		Statonary		FEDformer		Autoformer	
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
ETTm1	96	**0.316**	**0.347**	0.342	0.378	**0.316**	0.364	0.338	0.375	0.366	0.400	0.345	0.372	0.374	0.400	0.348	0.367	0.375	0.398	0.386	0.398	0.764	0.416	0.505	0.475
	192	**0.363**	**0.370**	0.372	0.393	**0.363**	0.390	0.371	0.387	0.396	0.414	0.380	0.389	0.400	0.407	0.387	0.385	0.408	0.410	0.459	0.444	0.426	0.441	0.553	0.496
	336	**0.392**	**0.390**	0.402	0.413	0.408	0.426	0.410	0.411	0.439	0.443	0.413	0.413	0.438	0.438	0.418	0.405	0.435	0.428	0.495	0.464	0.445	0.459	0.621	0.537
	720	**0.458**	**0.425**	0.462	0.449	0.459	0.464	0.478	0.450	0.540	0.509	0.474	0453	0.527	0.502	0.479	0.440	0.499	0.462	0.585	0.516	0.543	0.490	0.671	0.561
	avg	**0.383**	**0.384**	0.395	0.408	0.387	0.411	0.400	0.406	0.435	0.417	0.403	0.407	0.435	0.437	0.408	0.399	0.429	0.425	0.481	0.456	0.448	0.452	0.588	0.517
ETTm2	96	**0.169**	**0.248**	0.176	0.258	0.179	0.275	0.187	0.267	0.273	0.346	0.193	0.292	0.209	0.308	0.183	0.266	0.189	0.280	0.192	0.274	0.203	0.287	0.255	0.339
	192	**0.234**	**0.292**	0.244	0.304	0.262	0.326	0.249	0.309	0.350	0.421	0.284	0.362	0.311	0.382	0.247	0.305	0.253	0.319	0.459	0.444	0.269	0.328	0.281	0.340
	336	**0.294**	**0.339**	0.304	0.342	0.305	0.353	0.321	0.351	0.474	0.505	0.369	0.427	0.442	0.466	0.309	0.343	0.314	0.357	0.334	0.361	0.325	0.366	0.339	0.372
	720	0.390	**0.388**	0.408	0.403	**0.389**	0.407	0.497	0.403	1.347	0.812	0.554	0.522	0.675	0.587	0.407	0.398	0.414	0.413	0.417	0.413	0.421	0.415	0.433	0.432
	avg	**0.272**	**0.317**	0.283	0.327	0.284	0.340	0.291	0.333	0.609	0.521	0.350	0.401	0.409	0.436	0.287	0.328	0.292	0.342	0.306	0.347	0.305	0.349	0.327	0.371
ETTh1	96	0.383	**0.391**	0.426	0.426	0.398	0.427	0.384	0.402	0.391	0.417	0.386	0.400	0.424	0.432	0.388	0.401	0.494	0.479	0.513	0.419	**0.376**	0.419	0.449	0.459
	192	0.435	**0.420**	0.469	0.452	0.430	0.453	0.436	0.429	0.449	0.452	0.437	0.432	0.475	0.462	0.443	0.439	0.538	0.504	0.534	0.504	**0.420**	0.448	0.500	0.482
	336	0.479	**0.442**	0.506	0.473	**0.440**	0.460	0.491	0.469	0.510	0.489	0.481	0.459	0.518	0.521	0.488	0.466	0.574	0.521	0.588	0.535	0.459	0.465	0.521	0.496
	720	**0.471**	**0.461**	0.504	0.495	0.491	0.509	0.521	0.500	0.594	0.567	0.519	0.516	0.547	0.533	0.525	0.519	0.562	0.535	0.643	0.616	0.506	0.507	0.514	0.512
	avg	0.442	**0.429**	0.455	0.444	**0.440**	0.462	0.458	0.450	0.486	0.481	0.456	0.452	0.491	0.479	0.461	0.456	0.452	0.510	0.570	0.537	**0.440**	0.460	0.496	0.487
ETTh2	96	**0.281**	**0.330**	0.292	0.342	0.299	0.364	0.340	0.374	0.641	0.549	0.333	0.387	0.397	0.437	0.296	0.344	0.340	0.391	0.513	0.419	0.358	0.397	0.346	0.388
	192	**0.363**	**0.381**	0.387	0.400	0.422	0.441	0.402	0.414	0.896	0.656	0.477	0.476	0.520	0.504	0.389	0.402	0.430	0.439	0.512	0.493	0.429	0.439	0.456	0.452
	336	**0.411**	**0.418**	0.426	0.434	0.447	0.474	0.452	0.452	0.936	0.690	0.594	0.541	0.626	0.559	0.418	0.430	0.485	0.497	0.552	0.551	0.496	0.487	0.482	0.486
	720	**0.416**	**0.431**	0.430	0.446	0.442	0.467	0.462	0.468	1.390	0.863	0.831	0.657	0.863	0.672	0.433	0.448	0.500	0.497	0.562	0.560	0.463	0.474	0.515	0.511
	avg	**0.368**	**0.390**	0.384	0.406	0.402	0.437	0.414	0.427	0.966	0.690	0.559	0.515	0.602	0.543	0.384	0.406	0.439	0.452	0.526	0.516	0.437	0.449	0.450	0.459
Weather	96	**0.150**	**0.188**	0.176	0.218	0.161	0.229	0.172	0.220	0.164	0.232	0.196	0.255	0.182	0.242	0.193	0.234	0.237	0.312	0.173	0.223	0.217	0.296	0.266	0.336
	192	**0.202**	**0.238**	0.223	0.259	0.220	0.281	0.219	0.261	0.211	0.276	0.237	0.296	0.227	0.287	0.236	0.269	0.237	0.213	0.245	0.285	0.276	0.336	0.307	0.367
	336	**0.260**	**0.282**	0.277	0.297	0.278	0.331	0.280	0.306	0.269	0.327	0.283	0.335	0.282	0.334	0.288	0.304	0.298	0.353	0.321	0.338	0.339	0.380	0.359	0.395
	720	0.343	**0.353**	0.353	0.347	**0.311**	0.356	0.365	0.359	0.355	0.404	0.345	0.381	0.352	0.386	0.358	0.350	0.352	0.388	0.414	0.410	0.403	0.428	0.419	0.428
	avg	**0.239**	**0.261**	0.257	0.280	0.243	0.299	0.259	0.287	0.250	0.310	0.265	0.317	0.261	0.312	0.269	0.339	0.271	0.334	0.288	0.314	0.309	0.360	0.419	0.428
Electricity	96	**0.141**	**0.233**	0.190	0.296	0.164	0.269	0.168	0.272	0.254	0.347	0.197	0.282	0.207	0.307	0.198	0.276	0.187	0.304	0.169	0.273	0.193	0.308	0.201	0.317
	192	**0.160**	**0.250**	0.199	0.304	0.177	0.285	0.184	0.289	0.261	0.353	0.196	0.285	0.213	0.316	0.198	0.279	0.199	0.315	0.182	0.286	0.201	0.315	0.222	0.334
	336	**0.173**	**0.263**	0.217	0.319	0.193	0.304	0.198	0.300	0.273	0.364	0.209	0.301	0.230	0.333	0.217	0.301	0.212	0.329	0.200	0.304	0.214	0.329	0.254	0.361
	720	**0.197**	**0.284**	0.258	0.352	0.212	0.321	0.220	0.320	0.303	0.388	0.245	0.333	0.265	0.360	0.279	0.357	0.233	0.345	0.222	0.321	0.246	0.355	0.254	0.361
	avg	**0.168**	**0.258**	0.216	0.318	0.187	0.295	0.192	0.295	0.273	0.363	0.212	0.300	0.229	0.329	0.223	0.303	0.208	0.323	0.193	0.296	0.214	0.327	0.227	0.338
Traffic	96	**0.419**	**0.269**	0.462	0.315	0.519	0.309	0.593	0.321	0.558	0.320	0.650	0.396	0.615	0.391	0.649	0.391	0.607	0.392	0.612	0.338	0.587	0.366	0.613	0.388
	192	**0.443**	**0.276**	0.473	0.321	0.537	0.315	0.617	0.336	0.569	0.321	0.650	0.396	0.601	0.382	0.603	0.366	0.621	0.399	0.613	0.340	0.604	0.373	0.616	0.382
	336	**0.460**	**0.283**	0.494	0.331	0.534	0.313	0.629	0.336	0.591	0.328	0.605	0.373	0.613	0.386	0.613	0.371	0.622	0.396	0.618	0.328	0.621	0.383	0.622	0.337
	720	**0.490**	**0.299**	0.522	0.342	0.577	0.325	0.640	0.350	0.652	0.359	0.650	0.396	0.658	0.407	0.692	0.427	0.622	0.396	0.653	0.355	0.626	0.382	0.660	0.408
	avg	**0.453**	**0.282**	0.488	0.327	0.542	0.316	0.620	0.336	0.593	0.332	0.625	0.383	0.622	0.392	0.639	0.389	0.621	0.396	0.624	0.340	0.610	0.376	0.628	0.379

5.2 M4 Short Term Forecasting

M4 dataset (Makridakis et al., 2018) consists 100k time series. It covers time sequence data in various domains, including business, financial, and economy, and the sampling frequencies range from hourly to yearly. A table with summary statistics is presented in Appendix B, showing wide variability in time series characteristics. We follow the test setting suggested in (Wu et al., 2023) and fix the lookback length to be 2 times of forecasting length, and results are measured by Symmetric Mean Absolute Percentage Error (SMAPE), Mean Absolute Scaled Error (MASE) and Overall Weighted Average (OWA). We benchmark our model with N-BEATS (Oreshkin et al., 2020), N-HiTS (Challu et al., 2022), Informer (Zhou et al., 2021) and 9 baselines in long-term forecasting.

The results are summarized in Table 2. Our proposed model consistently outperforms benchmarks in all tasks. Specifically, we outperform the state-of-the-art MLP-based method N-BEATS (Oreshkin et al., 2020) by **1.8%** in SMAPE reduction. We also outperform the best Transformer-based method PatchTST (Nie et al., 2023) and the best CNN-based method TimesNet (Wu et al., 2023) by **1.5%** and **2.2%** in SMAPE reductions respectively. Since the M4 dataset only contains univariate time series, the attention to channels in our model plays a very limited role here. Thus good numerical performance indicates our dual transformer design with attention to hidden dimensions is also effective in univariate time series scenarios and can significantly boost forecasting performance.

Table 2: Short-term Forecasting tasks on M4 dataset. The best model is in boldface and the second best is underlined.

Models		CARD	PatchTST	MCIN	TimesNet	N-HiTS	N-BEATS	ETS.	LightTS	Dlinear	FEDformer	Stationary	Autoformer	Informer
Yearly	SMAPE	**13.214**	13.258	14.935	13.387	13.418	13.436	18.009	14.247	16.965	13.728	13.717	13.974	14.727
	MASE	**2.956**	2.985	3.523	2.996	3.045	3.043	4.487	3.109	4.283	3.048	3.078	3.134	3.418
	OWA	**0.776**	0.781	0.900	0.786	0.793	0.794	1.115	0.827	1.058	0.803	0.807	0.822	0.881
Quarterly	SMAPE	**9.961**	10.179	11.452	10.100	10.202	10.124	13.376	11.364	12.145	10.792	10.958	11.338	11.360
	MASE	**1.162**	1.212	1.389	1.182	1.194	1.169	1.906	1.328	1.520	1.283	1.325	1.365	1.401
	OWA	**0.876**	0.904	1.026	0.890	0.899	0.886	1.302	1.000	1.106	0.958	0.981	1.012	1.027
Monthly	SMAPE	**12.467**	12.641	13.773	12.670	12.791	12.667	14.588	14.014	13.514	14.260	13.917	13.958	14.062
	MASE	**0.914**	0.930	1.076	0.933	0.969	0.937	1.368	1.053	1.037	1.102	1.097	1.103	1.141
	OWA	**0.862**	0.876	0.983	0.878	0.899	0.880	1.149	0.981	0.956	1.012	0.998	1.002	1.024
Others	SMAPE	**4.478**	4.946	6.716	4.891	5.061	4.925	7.267	15.880	6.709	4.954	6.302	5.458	24.460
	MASE	**2.956**	2.985	4.717	3.302	3.216	3.391	5.240	11.434	4.953	3.264	4.064	3.865	20.960
	OWA	**0.959**	1.044	1.451	1.035	1.040	1.053	1.591	3.474	1.487	1.036	1.304	1.187	5.879
Avg	SMAPE	**11.638**	11.807	13.130	11.829	11.927	11.851	14.718	13.252	13.639	12.840	12.780	12.909	14.086
	MASE	**1.552**	1.590	1.896	1.585	1.613	1.599	2.408	2.111	2.095	1.701	1.756	1.771	2.718
	OWA	**0.835**	0.851	0.980	0.851	0.861	0.855	1.172	1.051	1.051	0.918	0.930	0.939	1.230

5.3 Boosting Effect of Signal Decay-based Loss Function

In this section, we present the boosting effect of our proposed signal decay-based loss function. In contrast to the widely used MSE loss function employed in previous training of long-term sequence forecasting models, our approach yields a reduction in MSE ranging from **3%** to **12%** across a spectrum of recent state-of-the-art baseline models, including transformer, convolutional, and MLP architectures as shown in Table 3. Our proposed loss function specifically empowers FEDformer and Autoformer, two algorithms that heavily rely on frequency domain information. This aligns with our signal decay paradigm, which acknowledges that frequency information carries variance/noise across time horizons. Our novel loss function can be considered a preferred choice for this task, owing to its superior performance compared to the plain MSE loss function.

Table 3: Influence for signal decay-based loss function. The lookback length is set as 96. All models are evaluated on 4 different predication lengths $\{96, 192, 336, 720\}$. The model name with * uses the robust loss proposed in this work. The better results are in boldface.

Models		CARD		CARD*		MCIN-regre		MCIN-regre*		TimesNet		TimesNet*		FEDformer		FEDformer*		Autoformer		Autoformer*	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.329	0.364	**0.316**	**0.347**	0.316	0.362	**0.313**	**0.350**	0.338	0.375	**0.321**	**0.356**	0.379	0.419	**0.344**	**0.380**	0.505	0.475	**0.450**	**0.442**
	192	0.368	0.385	**0.363**	**0.370**	0.363	0.390	**0.359**	**0.372**	**0.374**	0.387	0.377	**0.385**	0.426	0.441	**0.390**	**0.404**	0.553	0.537	**0.540**	**0.477**
	336	0.400	0.405	**0.393**	**0.390**	0.408	0.426	**0.392**	**0.399**	0.410	0.411	**0.401**	**0.400**	0.445	0.459	**0.436**	**0..433**	0.621	0.537	**0.594**	**0.505**
	720	0.468	0.444	**0.458**	**0.426**	0.481	0.476	**0.466**	**0.451**	0.478	0.450	**0.470**	**0.437**	0.543	0.490	**0.480**	**0.461**	0.671	0.561	**0.507**	**0.476**
	avg	0.391	0.400	**0.383**	**0.384**	0.392	0.414	**0.383**	**0.393**	0.400	0.406	**0.392**	**0.395**	0.448	0.452	**0.413**	**0.415**	0.588	0.528	**0.523**	**0.475**
ETTh1	96	0.387	0.399	**0.383**	**0.391**	0.421	0.431	**0.403**	**0.412**	**0.384**	0.402	0.389	**0.400**	0.376	0.419	**0.371**	**0.400**	**0.449**	0.459	0.453	**0.445**
	192	0.438	0.431	**0.435**	**0.420**	0.474	0.487	**0.471**	**0.451**	**0.436**	0.429	**0.436**	**0.425**	0.420	0.448	**0.419**	**0.432**	**0.500**	**0.482**	0.544	0.493
	336	0.486	0.454	**0.479**	**0.461**	0.569	0.551	**0.513**	**0.496**	0.491	0.469	**0.475**	**0.450**	0.459	0.465	**0.461**	**0.455**	**0.521**	0.496	0.535	**0.491**
	720	0.480	0.472	**0.471**	**0.429**	0.770	0.672	**0.720**	**0.636**	0.521	0.500	**0.494**	**0.477**	0.506	0.507	**0.491**	**0.482**	**0.514**	0.512	0.524	**0.495**
	avg	0.448	0.439	**0.442**	**0.425**	0.559	0.535	**0.527**	**0.499**	0.458	0.450	**0.449**	**0.438**	0.440	0.460	**0.436**	**0.442**	**0.496**	0.487	0.514	**0.481**

5.4 Are the Self-attention Scheme Effective for Long-term Forecasting?

The effectiveness of the self-attention scheme is questioned in (Zeng et al., 2023). They show that a linear layer can be used as a substitute for the self-attention layer to achieve higher accuracy in a transformer-based model, casting doubt on the efficacy of self-attention. However, we contend that this is not an inherent weakness of the self-attention scheme. Upon replacing channel-branch attention and temporal attention with a linear layer in CARD, we observe a consistent decline in accuracy across all datasets, as illustrated in Table 4. The deterioration effect is particularly pronounced in the weather dataset, which contains more informative covariates, with a significant drop of over **13%**. These findings suggest that the self-attention scheme may be more effective in feature extraction than a simple linear layer for time series forecasting.

5.5 Influence of Input Sequence Length

Previous research (Zeng et al., 2023; Wen et al., 2023) has highlighted a critical issue with the existing long-term forecasting transformers. They struggle to leverage extended input sequences, resulting in a decline in performance as the input length increases. In contrast, MLP-based models (Zeng et al., 2023) have demonstrated an ability to leverage longer input sequences to improve performance. We

Table 4: The effectiveness of the self-attention scheme. The lookback length is set as 96. CARD(tMLP) uses an MLP layer to substitute the token attention layer in CARD, CARD(cMLP) uses an MLP layer to substitute the channel attention layer in CARD, CARD(dMLP) uses two MLP layers to substitute both token and channel attention, and CARD(oMLP) contains only the embedding layer and an MLP layer.

Models		CARD		CARD(tMLP)		CARD(cMLP)		CARD(dMLP)		CARD(oMLP)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.316**	**0.347**	0.333	0.369	0.324	0.357	0.355	0.376	0.356	0.376
	192	**0.363**	**0.370**	0.375	0.390	0.371	0.381	0.393	0.394	0.393	0.394
	336	**0.393**	**0.390**	0.405	0.409	0.403	0.402	0.425	0.415	0.424	0.414
	720	**0.458**	**0.426**	0.467	0.444	0.463	0.436	0.489	0.451	0.467	0.444
	avg	**0.383**	**0.384**	0.395	0.403	0.390	0.394	0.415	0.409	0.415	0.408
Weather	96	**0.150**	**0.188**	0.160	0.207	0.172	0.213	0.195	0.234	0.195	0.234
	192	**0.202**	**0.238**	0.211	0.254	0.220	0.255	0.240	0.270	0.240	0.270
	336	**0.260**	**0.282**	0.270	0.296	0.276	0.296	0.292	0.306	0.292	0.306
	720	**0.343**	**0.335**	0.358	0.351	0.353	0.346	0.364	0.353	0.364	0.353
	avg	**0.239**	**0.261**	0.250	0.277	0.255	0.277	0.272	0.291	0.273	0.291

assert that this is not an inherent drawback of transformers, and CARD demonstrates robustness in handling longer and noisier historical sequence inputs, as evidenced by an **8.6%** and **8.9%** reduction in MSE achieved in the ETTh1 and ETTm1 datasets, respectively, when input lengths were extended from 96 to 720, as shown in Table 5.

5.6 Other Experiments

Table 5: Influence of prolonging input sequence. The lookback length is set as 96,192,336,720: CARD(96) means using lookback length 96.

Models		CARD(96)		CARD(192)		CARD(336)		CARD(720)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.384	0.391	0.378	0.390	0.372	**0.390**	**0.368**	0.392
	192	0.436	0.421	0.427	0.418	0.413	0.416	**0.407**	**0.416**
	336	0.479	0.443	0.458	0.434	0.437	0.431	**0.428**	**0.430**
	720	0.474	0.463	0.452	0.456	0.436	0.453	**0.418**	**0.449**
	avg	0.443	0.430	0.429	0.425	0.415	0.422	**0.405**	**0.421**
ETTm1	96	0.316	0.347	0.296	0.333	0.284	0.328	**0.288**	**0.332**
	192	0.363	0.370	0.342	0.359	0.326	0.354	**0.332**	**0.357**
	336	0.393	0.390	0.375	0.379	0.368	0.377	**0.364**	**0.376**
	720	0.458	0.426	0.439	0.418	0.428	0.410	**0.414**	**0.407**
	avg	0.383	0.384	0.363	0.372	0.352	0.367	**0.349**	**0.368**

We conducted a series of experiments, using both ablation and architecture variants, to evaluate each component in our proposed model. Our findings revealed that the channel branch made the greatest contribution to the reduction of MSE errors. The detailed results can be found in Appendix J.2. Furthermore, our experiments on sequential/parallel attention mixing design, detailed in Appendix J.1, show that our model design is the preferred option. Visual aids in the form of visualization graphs and attention maps can be found in Appendix A and H, which effectively demonstrate our accurate predictions and utilization of covariate information. Another noteworthy experiment, concerning the impact of training data size, is presented in Appendix K.1. This study revealed that using 70% of training samples can significantly improve performance for the 3/7 datasets affected by distribution shifts. Besides, Appendix F presents an error bar statistics table that demonstrates the robustness and small variance of CARD. Finally, more experiments and discussions are provided in Appendix due to space limitations.

6 Conclusion and Future Works

Conclusion In this paper, we present a novel dual transformer model, CARD, for time series forecasting. CARD is a Channel Dependent designed model that aligns information across different variables and hidden dimensions effectively. CARD improves traditional transformers by applying attention to both tokens and channels. The new design of the attention mechanism helps explore local information within each token, making it more effective for time series forecasting. Furthermore, we introduce a robust loss function to alleviate the issue of overfitting noises, an important issue in time series analysis. As demonstrated through various numerical benchmarks, our proposed model outperforms state-of-the-art models.

Future Works Our current model does not leverage the multi-scale representation of time series data, a process known to extract different layers of information from various levels of granularity, and has demonstrated its effectiveness for time series forecasting in several studies (e.g., Wang et al. 2023b). Additionally, it would be interesting to experiment with the proposed method for a wide range of tasks such as classification, anomaly detection, and imputation, as studied in Wu et al. (2023).

References

Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al. Xcit: Cross-covariance image transformers. *Advances in neural information processing systems*, 34:20014–20027, 2021.

Bao, H., Dong, L., Piao, S., and Wei, F. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

Challu, C., Olivares, K. G., Oreshkin, B. N., Garza, F., Mergenthaler-Canseco, M., and Dubrawski, A. N-hits: Neural hierarchical interpolation for time series forecasting. *arXiv preprint arXiv:2201.12886*, 2022.

Chen, S.-A., Li, C.-L., Yoder, N., Arik, S. O., and Pfister, T. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*, 2023.

Chen, W., Xing, X., Xu, X., Pang, J., and Du, L. Speechformer: A hierarchical efficient framework incorporating the characteristics of speech. *arXiv preprint arXiv:2203.03812*, 2022.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Das, A., Kong, W., Leach, A., Sen, R., and Yu, R. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ding, M., Xiao, B., Codella, N., Luo, P., Wang, J., and Yuan, L. Davit: Dual attention vision transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pp. 74–92. Springer, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gu, A., Goel, K., and Re, C. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Han, L., Ye, H.-J., and Zhan, D.-C. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *arXiv preprint arXiv:2304.05206*, 2023.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. arXiv: 1412.6980.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., and Yan, X. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Li, Z., Rao, Z., Pan, L., and Xu, Z. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. *arXiv preprint arXiv:2302.04501*, 2023.

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 2021.

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., and Dustdar, S. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022a.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, 2022b.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Ma, X., Zhou, C., Kong, X., He, J., Gui, L., Neubig, G., May, J., and Zettlemoyer, L. Mega: Moving average equipped gated attention. In *The Eleventh International Conference on Learning Representations*, 2023.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020.

PeMS. Traffic. URL `http://pems.dot.ca.gov/`.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022.

Rangapuram, S. S., Seeger, M., Gasthaus, J., Stella, L., Wang, Y., and Januschowski, T. Deep state space models for time series forecasting. In *Proceedings of the 32nd international conference on neural information processing systems*, pp. 7796–7805, 2018.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Sen, R., Yu, H.-F., and Dhillon, I. S. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32, 2019.

Smyl, S. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020.

UCI. Electricity. URL `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Wang, C., Chen, S., Wu, Y., Zhang, Z., Zhou, L., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023a.

Wang, H., Peng, J., Huang, F., Wang, J., Chen, J., and Xiao, Y. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023b.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. Transformers in time series: A survey. In *International Joint Conference on Artificial Intelligence(IJCAI)*, 2023.

Wen, R., Torkkola, K., Narayanaswamy, B., and Madeka, D. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

Wetterstation. Weather. URL `https://www.bgc-jena.mpg.de/wetter/`.

Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. Deeptime: Deep time-index meta-learning for non-stationary time-series forecasting. *arXiv preprint arXiv:2207.06046*, 2022a.

Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022b.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 101–112, 2021.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? 2023.

Zhang, T., Zhang, Y., Cao, W., Bian, J., Yi, X., Zheng, S., and Li, J. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022.

Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021.

Zhou, T., Ma, Z., Wang, X., Wen, Q., Sun, L., Yao, T., Yin, W., and Jin, R. FiLM: Frequency improved legendre memory model for long-term time series forecasting. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022a.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022b.

Zhu, C., Ping, W., Xiao, C., Shoeybi, M., Goldstein, T., Anandkumar, A., and Catanzaro, B. Long-short transformer: Efficient transformers for language and vision. *Advances in Neural Information Processing Systems*, 34:17723–17736, 2021.

A Visualization

B Datasets

Datasets of Long-term Forecasting Table 6 summarizes details of statistics of long-term forecasting datasets.

Table 6: Dataset details in long-term forecasting.

Dataset	Length	Dimension	Frequency
ETTm1	69680	7	15 min
ETTm2	69680	7	15 min
ETTh1	17420	7	1 hour
ETTh2	17420	7	1 hour
Weather	52696	22	10 min
Electricity	26304	321	1 hour
Traffic	17544	862	1 hour

M4 datasets of short-term Forecasting Table 7 summarizes details of statistics of short-term forecasting M4 datasets.

Table 7: Datasets and mapping details of M4 dataset.

Dataset	Length	Horizon
M4 Yearly	23000	6
M4 Quarterly	24000	8
M4 Monthly	48000	18
M4 Weekly	359	13
M4 Daily	4227	14
M4 Hourly	414	48

C Model Configuration

For all experiments, we use Adam optimizer (Kingma & Ba, 2017) with cosine learning rate decay after linear warm-up. The details are summarized in Table 8.

Table 8: Model configurations.

Dataset	encoder	patch	stride	model dim	ffn dim	heads dim	dp dim	dropout	learning rate	train epoch	warm-up	batch size
ETTm1	2	16	8	16	32	8	8	0.3	1e-4	100	0	128
ETTm2	2	16	8	16	32	8	8	0.3	1e-4	100	0	128
ETTh1	2	16	8	16	32	8	8	0.3	1e-4	100	0	128
ETTh2	2	16	8	16	32	8	8	0.3	1e-4	100	0	128
Weather	2	16	8	128	256	8	8	0.2	1e-4	100	0	128
Electricity	2	16	8	128	256	8	8	0.2	1e-4	100	20	32
Traffic	2	16	8	128	256	8	8	0.2	1e-4	100	20	24
M4 Hourly	2	16	8	128	256	8	8	0.2	5e-4	100	0	128
M4 Weekly	2	16	8	128	256	8	8	0.2	5e-4	100	0	128
M4 Daily	2	16	8	128	256	8	8	0.2	5e-4	100	0	128
M4 Monthly	2	16	8	128	256	8	8	0.2	5e-4	100	0	128
M4 Quarterly	2	4	2	128	256	8	8	0.2	5e-4	100	0	128
M4 Yearly	2	3	1	128	256	8	8	0.2	5e-4	100	0	128

D Sample Code

E Experiments for All Benchmarks Datasets with Prolonging Input Sequence

In this section, we report the proposed model with 720 input length. For each benchmark, we report the best results in the literature or conduct grid searches on input length to build strong baselines.

F Error Bar Statistics

G Training Speed for Different Input Sequence Length

H Attention Pattern Maps

I Related Works

Patchfied Transformers in other Domains Transformer (Vaswani et al., 2017) has demonstrated significant potential in different data modalities. Among all applications, patching is an essential part when local semantic information is important. In NLP, BERT (Devlin et al., 2018), GPT (Radford et al., 2019) and their follow-up models consider subword-based tokenization and outperform character-based tokenization. In CV, Vision Transformers (e.g., Dosovitskiy et al. 2020; Liu et al. 2021; Bao et al. 2022; Ding et al. 2022; He et al. 2022) split an image into patches and then feed into the Transformer models. Similarly, in speech fields, researchers use convolutions to extract information in sub-sequence levels from a raw audio input (e.g., Hsu et al. 2021; Radford et al. 2022; Chen et al. 2022; Wang et al. 2023a).

J Others

J.1 Architecture Variants

The present study encompasses the design of five distinct sequential and parallel feature flow architectures, with the aim of integrating both temporal signal and channel-aligned information, as depicted in Figure 9. Following an exhaustive analysis, it is concluded that the architecture featuring the channel branch, complemented by channel/time fusion, is the most resilient variant. Consequently, this specific architecture is adopted as the default approach in this work.

Table 9: Model variants. All models are evaluated on 4 different predication lengths $\{96, 192, 336, 720\}$. The best results are in boldface.

Models		c->t+c(CARD)		t->c+t		t+c		t->c		c->t	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.316**	**0.347**	0.318	0.346	0.318	0.346	0.326	0.363	0.334	0.368
	192	**0.363**	**0.370**	0.367	0.370	0.366	0.369	0.366	0.385	0.372	0.387
	336	**0.393**	**0.390**	0.399	0.391	0.396	0.391	0.400	0.404	0.401	0.407
	720	**0.458**	**0.426**	0.466	0.429	0.463	0.428	0.459	0.440	0.458	0.438
	avg	**0.383**	**0.384**	0.388	0.384	0.386	0.384	0.388	0.398	0.391	0.400
Weather	96	**0.150**	**0.188**	0.153	0.193	0.152	0.189	0.152	0.191	0.152	0.192
	192	0.202	0.238	0.203	0.239	**0.201**	**0.236**	**0.201**	0.239	0.203	0.240
	336	**0.260**	0.282	0.269	0.288	0.261	**0.281**	0.263	0.284	0.262	0.284
	720	**0.343**	**0.335**	0.345	0.339	0.344	0.337	0.347	0.339	0.344	0.337
	avg	**0.239**	**0.261**	0.243	0.265	0.240	**0.261**	0.241	0.263	0.240	0.263

J.2 Component Ablation Experiments

We conducted a series of with/without ablation experiments on each component in our proposed model. Consistent with our design, as shown in table 10, the channel branch exhibited the greatest contribution to the reduction of mean squared error (MSE); its removal resulted in a 2% and 7% increase in MSE for ETTm1 and Weather, respectively. The dual attention time branch contributed approximately 1% to the reduction of MSE. The dynamics projection did not significantly contribute to accuracy improvements but did provide efficiencies. Furthermore, the position embedding was deemed unnecessary for our model, as the patchwise design adequately utilized temporal information.

Table 10: Component Ablation Experiments by removing dynamic (dynamic projection), smooth (EMA), dual (attention over hidden dimension), channel(dual transformer over channel dimension), and embed (positional embedding) sequentially. All models are evaluated on 4 different predication lengths $\{96, 192, 336, 720\}$. The differences in thousandths w.r.t. predecessor models are reported in parentheses.

Models		CARD		wo. dynamic		wo. smooth		wo. dual		wo. channel		wo. embed	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.316	0.347	0.313 (+3)	0.344 (+3)	0.322 (-9)	0.346 (-2)	0.322 (0)	0.345 (+1)	0.326 (-4)	0.348 (-3)	0.326 (0)	0.348 (0)
	192	0.363	0.370	0.361 (+2)	0.368 (+2)	0.363 (-2)	0.370 (-2)	0.364 (-1)	0.370 (0)	0.372 (-8)	0.370 (0)	0.372 (0)	0.371 (-1)
	336	0.393	0.390	0.393 (0)	0.389 (+1)	0.393 (0)	0.389 (0)	0.395 (-2)	0.391 (-2)	0.404 (-9)	0.393 (-2)	0.404 (0)	0.394 (-1)
	720	0.458	0.426	0.462 (-4)	0.426 (0)	0.458 (+4)	0.425 (+1)	0.462 (-4)	0.427 (-2)	0.470 (-8)	0.429 (-2)	0.471 (0)	0.430 (0)
	avg	0.383	0.384	0.382 (0.3)	0.382 (1.5)	0.384 (-1.8)	0.382 (-0.8)	0.386 (-1.8)	0.383 (-0.8)	0.393 (-7.3)	0.408 (-1.8)	0.343 (-0.3)	0.386 (-0.8)
Weather	96	0.150	0.188	0.150 (0)	0.187 (+1)	0.151 (-1)	0.190 (-3)	0.151 (0)	0.191 (-1)	0.173 (-22)	0.205 (-14)	0.173 (0)	0.205 (0)
	192	0.202	0.238	0.198 (+2)	0.234 (+4)	0.201 (-3)	0.237 (-4)	0.201 (0)	0.236 (1)	0.220 (-19)	0.247 (-11)	0.220 (0)	0.247 (0)
	336	0.260	0.282	0.258 (2)	0.279 (3)	0.259 (-1)	0.280 (-1)	0.263 (-4)	0.282 (-2)	0.275 (-12)	0.287 (-5)	0.276 (-1)	0.288 (-1)
	720	0.343	0.335	0.339 (4)	0.334 (1)	0.342 (-3)	0.335 (-1)	0.341 (1)	0.335 (0)	0.354 (-14)	0.339 (-4)	0.355 (-1)	0.339 (0)
	avg	0.239	0.261	0.236 (2.5)	0.259 (2.2)	0.238 (-2)	0.261 (-2)	0.239 (-0.8)	0.261 (-0.5)	0.256 (-16.5)	0.270 (-8,5)	0.256 (-0.5)	0.270 (-0.3)

K More Analysis on Transformers for Time Series Forecasting

K.1 Is Training Data Size a Limiting Factor for Existing Long-Term Forecasting Transformers?

We have observed a distribution shift phenomenon in fifty percent of the benchmark datasets: Traffic, ETTh2, and ETTm2. The model's performance demonstrates a significant enhancement with the use of only 70% training data samples compared to the standard training setting for long-term forecasting, as illustrated in table 11. While it has been argued that the transformer model exhibits a weakness where more training data fails to improve performance Zeng et al. (2023), we contend that this issue is an inherent feature of each time series benchmark dataset, wherein changes in data distribution between historical and current data are not related to the transformer model. Nevertheless, further exploration of this phenomenon may lead to improved performance, and we thus leave it as a topic for future study.

Table 11: Less training data experiment.

Tasks		ETTm1		ETTm2		ETTh1		ETTh2		Weather		Electricity		Traffic	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
All samples	96	**0.316**	**0.347**	0.169	0.248	**0.383**	**0.391**	0.281	0.330	**0.150**	**0.188**	**0.141**	**0.233**	0.419	0.269
	192	**0.363**	**0.370**	0.234	0.292	**0.435**	**0.420**	0.363	0.381	**0.202**	**0.238**	**0.160**	**0.250**	0.443	0.276
	336	**0.393**	**0.390**	0.294	0.339	**0.479**	**0.442**	0.411	0.418	**0.260**	**0.282**	**0.173**	**0.263**	0.460	0.283
	720	**0.458**	**0.426**	0.390	0.388	**0.471**	**0.461**	0.416	0.431	**0.343**	**0.335**	**0.197**	**0.284**	0.453	0.282
	avg	**0.383**	**0.384**	0.272	0.317	**0.442**	**0.429**	0.368	0.390	**0.329**	**0.261**	**0.168**	**0.258**	0.453	0.282
70% Samples	96	0.350	0.431	**0.163**	**0.242**	0.425	0.431	**0.272**	**0.325**	0.245	0.263	0.157	0.239	**0.404**	**0.263**
	192	0.401	0.403	**0.225**	**0.285**	0.482	0.462	**0.350**	**0.374**	0.312	0.310	0.180	0.257	**0.428**	**0.273**
	336	0.440	0.428	**0.284**	**0.324**	0.528	0.485	**0.394**	**0.411**	0.382	0.352	0.197	0.270	**0.444**	**0.471**
	720	0.514	0.471	**0.371**	**0.378**	0.529	0.506	**0.403**	**0.427**	0.473	0.405	0.229	0.296	**0.471**	**0.296**
	avg	0.426	0.419	**0.261**	**0.307**	0.491	0.471	**0.355**	**0.384**	0.353	0.333	0.191	0.266	**0.437**	**0.278**