ALTTS: DECOUPLING AUTOREGRESSION AND CROSS-VARIABLE DEPENDENCY VIA ALTERNATING OPTIMIZATION FOR MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

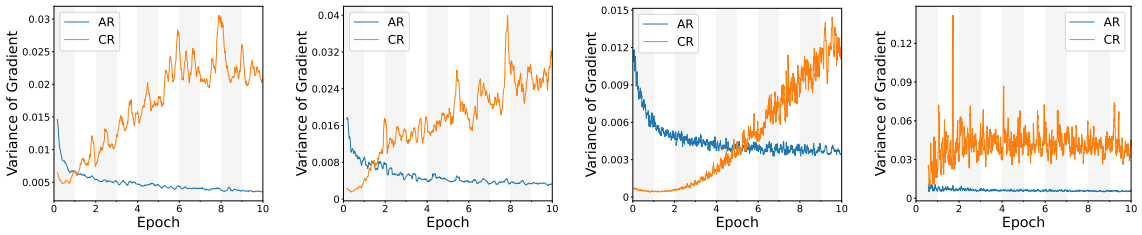
ABSTRACT

Multivariate time series mix two qualitatively heterogeneous components: (i) consistent autoregressive dependencies within individual series, and (ii) intermittent cross-dimension interactions that are often spurious over long horizons. Channel Dependent (CD) methods resolve the spatial complexity through sparse modeling or channel clustering, but the two components are modeled without distinction. We show that training a single structure to capture both effects poses challenges for optimization, as the high-variance updates required to model cross-dimension relations contaminate the gradients needed for autoregression, leading to brittle learning and degraded long-horizon accuracy. Motivated by this observation, we develop ALTTS, a dual-path framework that explicitly decouples autoregression and cross-relation modeling. In ALTTS, the autoregression path is realized by a linear predictor and the cross-relation path by a Transformer with Cross-Relation Self-Attention (CRSA), while the two are coordinated through alternating optimization to isolate gradient noise and reduce cross-block interference. Extensive experiments across multiple benchmarks demonstrate that ALTTS consistently outperforms existing models, particularly on long-horizon forecasting tasks. These results highlight that carefully designed optimization strategies, rather than increasingly complex architectures, can be the key to advancing multivariate time series forecasting.

1 INTRODUCTION

Time series forecasting aims to estimate future outcomes from past observations. Classical multivariate time series analysis rests on structural and probabilistic assumptions that enable tractable inference. Vector autoregression (VAR) offers a concise baseline for joint stationary dynamics (Sims, 1980), while cointegration theory captures co-movements among non-stationary series (Engle & Granger, 1987). Modern methodologies further provide systematic approaches for long-horizon, high-dimensional time series with diverse cross-time, cross-variable interactions (Newey & West, 1987; Bai & Ng, 2002; Basu & Michailidis, 2015).

Building on these foundations, recent deep learning methods extend time series modeling and have demonstrated significant scalability and strong inference ability for long-term time series forecasting (LTSF). Representative lines include CNN- and RNN-based (Lai et al., 2017; Li et al., 2018; LIU et al., 2022), graph-based (Yu et al., 2018; Shang et al., 2021), and Transformer-based (Zhou et al., 2021; 2022) models. These models tailor the network structure to the characteristics of time series data, among which techniques such as patching (Nie et al., 2023), sparse modeling (Lin et al., 2024) and dependency modeling (Zhang & Yan, 2023; Hu et al., 2025) have delivered notable gains across various real-world settings. At the same time, the recent success of MLP-based and linear models (Oreshkin et al., 2020; Zeng et al., 2023; Das et al., 2023;



(a) ETTh1 (b) ETTh2 (c) ETTm1 (d) ETTm2

Figure 1: Variance of gradient for AR and CR under joint training across four ETT datasets. Higher variance reflects greater training instability, motivating the use of alternating optimization in ALTTS.

Huang et al., 2025) call into question whether increasingly complex architectures are necessary for LTSF, and motivate re-examining how to represent the core properties of multivariate time series.

From an optimization perspective, training modern neural networks operates in a non-convex and often non-smooth regime. Foundational analyses establish the convergence of stochastic first-order methods, while adaptive approaches such as Adam (Kingma & Ba, 2015) have inspired refinements and provably convergent variants like AMSGrad (Reddi et al., 2018). For structured objectives, proximal alternating methods (Bolte et al., 2014) and block-coordinate descent (BCD) guarantee monotone descent and convergence, that are attracting growing interest within the deep learning community (Du et al., 2019; Zeng et al., 2019). More broadly, the integration of optimization principles with network design, exemplified by embedding differentiable optimization layers in networks (Amos & Kolter, 2017) and unfolding iterative algorithms for model alignment (Luo et al., 2020), has emerged as a powerful paradigm for taming ill-conditioned learning dynamics. In the context of LTSF, unstable learning dynamics can originate from heterogeneous dependency structures. Figure 1 reports the variance of gradient for autoregression (AR) and cross-relation (CR) parameters under the joint training schedule. Across all cases, the CR block exhibits substantially larger variance while the variance with the AR block remains comparatively low and decays over epochs.

Based on the above observations, we propose ALTTS, a dual-path framework that decouples dependency modeling and trains the two paths via alternating optimization. In ALTTS, AR and CR are separately forecasted by two modules. Subsequently, alternating optimization is applied to alleviate the entanglement of seemingly homogeneous AR and CR through cyclic block-wise updates. Experimentally, ALTTS achieves state-of-the-art performance in a wide range of time series forecasting tasks with minimal architectural sophistication. Our contributions are as follows:

1. We present ALTTS, the first deep learning framework that explicitly *decouples* autoregression and cross-variable dependency and coordinates them via *alternating optimization*. Our analysis further reveals the risk of gradient entanglement when these patterns are learned jointly, especially when the true longitudinal structure is unobservable.
2. We evaluate ALTTS on seven widely used multivariate time series benchmarks against strong linear, Transformer-based, and hybrid baselines. ALTTS consistently achieves competitive or superior performance across datasets and horizons while relying only on canonical architectures, underscoring the effectiveness of optimization-driven design.
3. Beyond empirical results, ALTTS highlights that *training schedules can be treated as a design variable*, pointing to new opportunities for integrating optimization principles into the structural design of neural networks.

2 Related Work

Long-Term Time Series Forecasting Transformer-based models have been a dominant thread in long-term time series forecasting (LTSF). Many advancements involve adapting the Transformer (Vaswani et al., 2017) for LTSF. These models utilize various properties of long-term time series through seasonal-trend decomposition (Wu et al., 2021; Wang et al., 2024), frequency analysis (Zhou et al., 2022; Chen et al., 2024), and sparse modeling (Zhou et al., 2021; Luo & Wang, 2024). Further investigations into the characteristics of time-series data show that simple models, such as linear layers (Zeng et al., 2023; Li et al., 2023), can achieve comparable or superior performance to Transformers. In parallel, there is a growing interest in dependency modeling. The Channel Independent (CI) methods rely solely on the historical information of each series, individually or through a pooled Channel Mix (CM) modeling, thus concentrating on autoregressive patterns. PatchTST (Nie et al., 2023) proposes a CI/CM patching, showing consistent gains on long horizons. Following this path, pure autoregressive models including TimesNet (Wu et al., 2023), SparseTSF (Lin et al., 2024), and TimeBase (Huang et al., 2025) achieve impressive performance, demonstrating the effectiveness of CI modeling. In contrast, the Channel Dependent (CD) methods explicitly target cross-dimension structures. Crossformer (Zhang & Yan, 2023) designs a two-stage attention to capture cross-time and cross-dimension dependencies. iTransformer (Liu et al., 2023) tokenizes variables and applies attention to their time embeddings, where attention scores implicitly represent multivariate correlations. To resolve the high variable dimensionality in large datasets, Channel Clustering (CC) is proposed, analogous to patching in time domain modeling. In CC methods, such as DUET (Qiu et al., 2025) and TimeFilter (Hu et al., 2025), heterogeneous variables are grouped together to preserve instantaneous correlations. The two-stage temporal-spatial paradigm is widely adopted in most CD methods. While recent studies also experiment on structures compatible with both CI and CD settings (Lin et al., 2025), further research on the integration of autoregressive and cross-dimension modeling is still needed.

Alternating Optimization Alternating optimization methods attract increasing interest in deep learning, such as Block-Coordinate Descent (BCD) and Alternating Direction Method of Multipliers (ADMM) by Boyd et al. (2010). Algorithms that allow general non-smooth and non-convex problems provide theoretical foundations for the application in deep learning scenarios (Razaviyayn et al., 2013; Bolte et al., 2014; Razaviyayn et al., 2014). Meanwhile, the alternating optimization algorithms are employed heuristically in adversarial learning (Goodfellow et al., 2014) and Computer Vision (Luo et al., 2020; Akbari et al., 2023), where two or more neural networks and objectives are alternately optimized for better alignment. In this paper, we explore the alternating optimization of dependency modeling schemes for LTSF to coordinate autoregressive and cross-dimension patterns while preserving temporal causality.

3 Methodology

3.1 General Structure

In the multivariate time series forecasting task, let $\mathbf{X}_t = [\mathbf{x}_t^{(1)}, \ldots, \mathbf{x}_t^{(D)}] \in \mathbb{R}^{D \times L}$ be the lookback window at step t, where each $\mathbf{x}_t^{(i)} = [x_{t-L+1}^{(i)}, \ldots, x_t^{(i)}]$ represents a historical input sequence of length L. D denotes the number of dimensions. The target sequence is $\mathbf{Y}_{t+1} = [\mathbf{y}_{t+1}^{(1)}, \ldots, \mathbf{y}_{t+1}^{(D)}] \in \mathbb{R}^{D \times H}$, where $\mathbf{y}_{t+1}^{(i)} = [x_{t+1}^{(i)}, \ldots, x_{t+H}^{(i)}]$ is the realization of the i-th sequence from step $t+1$ to $t+H$. For the individual sequences, we omit the subscript t and use $\mathbf{x}_i := \mathbf{x}_t^{(i)}$, $\mathbf{y}_i := \mathbf{y}_{t+1}^{(i)}$ for short in the following analysis.

In general, to generate the predicted series $\hat{\mathbf{Y}}_{t+1}$, a transition matrix is used. Let $\mathbf{F} = (f_{ij})_{D \times D}$ be a matrix of projections. Each projection $f_{ij} : \mathbb{R}^L \mapsto \mathbb{R}^H$ measures the contribution of $\mathbf{x}_j$ to $\mathbf{y}_i$. $\mathbf{F}$, f_{ij} can be approximated by a neural network or part of a neural network, denoted by $\hat{\mathbf{F}}$ and $\hat{f}_{ij}$ respectively. For

convenience, we define an "apply-then-sum" operator $*$ as

$$\mathbf{F} * \mathbf{X}_t := \left(\sum_{j=1}^{D} f_{1j}(\mathbf{x}_j), \ldots, \sum_{j=1}^{D} f_{Dj}(\mathbf{x}_j) \right) \in \mathbb{R}^{D \times H}. \tag{1}$$

Note that the $*$ operation is linear with respect to the first argument. The transition equation is therefore

$$\mathbf{F} * \mathbf{X}_t + \mathbf{V}_{t+1} = \mathbf{Y}_{t+1}, \tag{2}$$

where $\mathbf{V}_{t+1}$ represents the unpredictable innovations of the time series at step $t+1$ given $\mathbf{X}_t$. This general form summarizes CI, CM, and CD methods. Setting all off-diagonal projections to zero, $f_{ij} \equiv 0, i \neq j$ yields the CI methods, where D independent neural networks are used to model the autoregression per series. Further imposing $f_{11} = \cdots = f_{DD}$ gives the CM setting, where a single shared neural network generalizes the autoregressive patterns from all series. Allowing non-trivial off-diagonal entries, i.e., at least one $f_{ij}, i \neq j$ depends on the input, leads to the CD methods with explicit cross-dimension effects.

However, the fully dense specification is computationally inefficient and susceptible to overfitting, especially when each f_{ij} is independently parameterized. To better utilize the cross-variable structure, recent work adopts Channel Clustering (CC), which is equivalent to a block-diagonal constraint to $\mathbf{F}$. Essentially, the $D \times D$ entries of $\mathbf{F}$ are estimated by a significantly smaller number of modules, reflecting the low intrinsic dimensionality of many long-horizon high-dimensional time series. Through different grouping and coupling schemes for $\{f_{ij}\}$, one can instantiate different structural assumptions on inter-series dependence.

3.2 AR-CR Structural Decoupling

Modeling cross-dimension dependencies is challenging in multivariate forecasting. Autoregression (AR) is typically stable and persistent, whereas cross-relations (CR) are often regime-dependent and instantaneous. However, most dependency modeling methods either discard CR completely or homogeneously model the two. We instead propose a dual-path design that explicitly separates AR and CR modeling, as illustrated in Figure 2. Rather than estimating the full operator at once, we break down $\mathbf{F}$ into diagonal and off-diagonal components, $\mathbf{F} = \mathbf{F}_{\text{AR}} + \mathbf{F}_{\text{CR}}$, due to their distinct properties. $\mathbf{F}_{\text{AR}} = \text{diag}(f_{11}, \ldots, f_{DD})$ captures per-series AR patterns, and $\mathbf{F}_{\text{CR}} = \mathbf{F} - \mathbf{F}_{\text{AR}}$ encodes all cross-dimension dependencies.

Auto-Regression Path For each variable i, we first apply Reversible Instance Normalization, RevIN (Kim et al., 2021), and fit a linear predictor $\hat{f}_{ii}$

$$\hat{\mathbf{y}}_i^{\text{AR}} = \hat{f}_{ii}(\mathbf{x}_i) \tag{3}$$

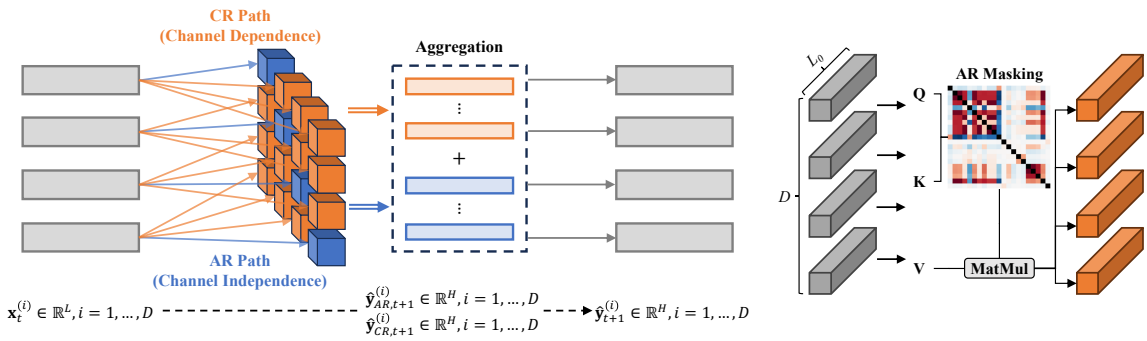
Parameters are not shared across i, enforcing channel independence. The AR path coincides with the RLinear model (Li et al., 2023). This simple model provides a clean separation of autoregression from cross-dimension effects. The AR path is trained with L1 regularization to prevent overfitting in long-horizon modeling.

Cross-Relation Path We apply an inverted Transformer encoder that attends across variables. To prevent leakage of AR information into the CR module, we mask intra-series links in the attention matrix. Each individually normalized time series is compressed into a L_0-dimensional token via a linear projection,

$$\mathbf{Z}_0 = \text{Embedding}(\mathbf{X}_t) \tag{4}$$

where $\mathbf{Z}_0 \in \mathbb{R}^{D \times L_0}$ denotes the temporal embeddings of variables. $\mathbf{Z}_0$ is passed into the Multi-Head Self-Attention (MHSA) layer (Vaswani et al., 2017). For each head, the Cross-Relation Self-Attention (CRSA) is calculated as

$$\text{CRSA}(\mathbf{Z}_0) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}} + \mathbf{M}\right)\mathbf{V} \tag{5}$$



(a) ALTTS Model Overview (b) Cross-Relation Self-Attention

Figure 2: Architecture of ALTTS. (a) Multivariate time series is passed into two parallel paths, the channel-independent AR path and the channel-dependent CR path. Outputs are summed to obtain the final prediction. (b) The cross-relation self-attention forms queries/keys/values from per-variable embeddings and an AR mask is applied to the attention matrix to suppress intra-series links.

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{D \times d_h}$ denote queries, keys, and values, respectively. The attention learns *cross-variable* rather than cross-time dependencies. As such, the attention matrix $\mathbf{A} = (\mathbf{Q}\mathbf{K}^\top / \sqrt{d_h})$ represents the multivariate correlations, same as iTransformer (Liu et al., 2023). The additional mask $\mathbf{M} = \mathrm{diag}(-\infty, \ldots, -\infty)$ zeros out diagonal attention weights in CRSA. This prevents the CR module from duplicating the AR function, thus enforcing a CR-only modeling. The remainder of the encoder follows the standard Transformer block Vaswani et al. (2017)

$$\begin{aligned} \mathbf{Z}_1 &= \mathrm{LayerNorm}(\mathbf{Z}_0 + \mathrm{MHSA}(\mathbf{Z}_0)) \\ \mathbf{Z}_2 &= \mathrm{LayerNorm}(\mathbf{Z}_1 + \mathrm{MLP}(\mathbf{Z}_1)) \end{aligned} \tag{6}$$

where MLP denotes a two-layer feedforward network. A channel-independent linear head maps the encoder output to the CR components $\hat{\mathbf{y}}_i^{\mathrm{CR}}$ in the target sequence.

The final output is the denormalized sum of the AR and CR outputs. Our implementation uses a parameter-free RevIN layer, so the two paths share no trainable parameters.

3.3 BLOCK ALTERNATING OPTIMIZATION

Gradient Entanglement of Joint AR-CR Training For simplicity, the following discussion considers the MSE loss function under minibatch optimization, so gradients are stochastic. With perfect knowledge of cross-variable contributions $f_{ij}, j = 1, \ldots, D$ for each variable i, the gradient for $\hat{f}_{ij}$ can be estimated as

$$\nabla_{\theta_{ij}} \mathcal{L}^* = -J_{ij}^\top \mathbf{r}_{ij}, \tag{7}$$

where $\mathcal{L}^* := \frac{1}{2} \sum_{i,j=1}^{D} \|\mathbf{r}_{ij}\|_2^2$. θ_{ij} is the set of parameters for $\hat{f}_{ij}$, J_{ij} is the Jacobian, and $\mathbf{r}_{ij} := f_{ij}(\mathbf{x}_j) - \hat{f}_{ij}(\mathbf{x}_j)$ is the residual of the fitted projection $\hat{f}_{ij}$. Because (7) is additively separable across i, j, the update is unaffected by other projections. The learned $\hat{f}_{ij}$ is therefore consistent with the true projection f_{ij}.

However, in real-world time series forecasting, such decomposition of residuals is infeasible. Without access to the true transition matrix $\mathbf{F}$, we only observe the aggregate residuals $\mathbf{r}_i := \mathbf{y}_i - \hat{\mathbf{y}}_i = \sum_{j=1}^{D} \mathbf{r}_{ij}, i = 1, \ldots, D$. The estimated gradient mixes estimation errors from multiple projections,

$$\nabla_{\theta_{ij}} \mathcal{L} = -J_{ij}^\top \mathbf{r}_i, \tag{8}$$

where $\mathcal{L} := \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{r}_i\|_2^2$. Minimizing the sum of $\|\mathbf{r}_i\|_2^2$ does not guarantee the consistency between each individual $\hat{f}_{ij}$ and f_{ij}, but only the combined mapping over j. Equation 8 induces gradient entanglement between parameter blocks θ_{ii} and θ_{ij} through the shared residual $\mathbf{r}_i$ throughout training. When series are relatively homogeneous and exhibit spurious correlations, this entanglement blurs the distinction between AR and CR, often leading to over-reliance on cross-variable information.

Let $\mathbf{r}_{-ii} := \sum_{j\neq i}^{D} \mathbf{r}_{ij}$ be the sum of CR residuals for variable i. For each AR projection $\hat{f}_{ii}$, the unbiasedness of (8) requires the conditional expectation of the bias term to be zero.

$$\mathbb{E}(-J_{ii}^{\top}\mathbf{r}_{-ii} \mid \mathbf{x}_i) = -J_{ii}^{\top}\mathbb{E}(\mathbf{r}_{-ii} \mid \mathbf{x}_i) = 0, \tag{9}$$

which requires $\mathbb{E}(\mathbf{r}_{-ii} \mid \mathbf{x}_i) = 0$. However, due to the existence of cross-variable dependencies and the shared residual $\mathbf{r}_i$ among $\hat{f}_{ij}$, this requirement fails to hold unless projections $\hat{f}_{ij}, j \neq i$ are constant with respect to $\mathbf{x}_i$ as in the CI setting. In the general CD setting, where $\mathbb{E}(\mathbf{x}_j \mid \mathbf{x}_i) \neq 0$ for some j, $\mathbb{E}(\mathbf{r}_{-ii} \mid \mathbf{x}_i)$ is a non-trivial function of $\mathbf{x}_i$.

The bias in the mixed gradient may not pose a problem for forecasting since unbiased predictions can still be attainable. Nonetheless, it can degrade training stability. Conditional on $\mathbf{x}_i$, the covariance of (7) is 0, whereas the covariance of the mixed gradient (8), $\mathrm{Cov}(-J_{ii}^{\top}\mathbf{r}_i \mid \mathbf{x}_i) = \mathrm{Cov}(-J_{ii}^{\top}\mathbf{r}_{-ii} \mid \mathbf{x}_i) \succeq 0$, is generally non-trivial. This additional covariance introduced by the CR contamination translates into noisier updates for the AR block. We consider the following assumption to investigate the unconditional covariance.

Assumption 1 *Let $\Sigma_{ii} := \mathrm{Cov}(-J_{ii}^{\top}\mathbf{r}_{ii})$, $\Sigma_{-ii} := \mathrm{Cov}(-J_{ii}^{\top}\mathbf{r}_{-ii})$ be the covariance matrices of the true gradient and the CR contamination respectively. Assume the covariances satisfy*

$$\|\Sigma_{-ii}\|_1 > 4\|\Sigma_{ii}\|_1, \tag{10}$$

where $\|\cdot\|_1$ is the trace norm.

The trace norm of gradient covariance serves as a metric for update turbulence, with the same definition as in Johnson & Zhang (2013); Agarwal et al. (2022). Equation 10 suggests a lower bound for $\|\Sigma_{-ii}\|_1$ when the CR gradient variance is sufficiently large to affect the optimization of the AR path under various settings.

When a subset of variables in the input sequence is highly correlated, the indistinguishability arises and can lead to equation 10. For instance, in a bivariate case with an instantaneous relationship $\mathbf{x}_i = \alpha\mathbf{x}_j + \epsilon$, $\forall t$ for some constant α while cross-dimension contributions are zero (i.e., $f_{ij} = f_{ji} = 0$), $\|\Sigma_{-ii}\|_1$ can be inflated by misspecified $\hat{f}_{ij}$. Even without extreme dependence, small cross-dimension errors that arise intermittently can accumulate and materially raise gradient variance over training. To assess its impact on the AR gradient stability, consider the conditional covariance matrix of (8)

$$\mathrm{Cov}(J_{ii}^{\top}\mathbf{r}_i) = \Sigma_{ii} + \Sigma_{-ii} + 2\mathrm{Cov}(J_{ii}^{\top}\mathbf{r}_{ii}, J_{ii}^{\top}\mathbf{r}_{-ii}). \tag{11}$$

The trace norm of equation 11 satisfies

$$\begin{aligned}\|\mathrm{Cov}(J_{ii}^{\top}\mathbf{r}_i)\|_1 &= \|\Sigma_{ii}\|_1 + \|\Sigma_{-ii}\|_1 + 2\,\mathrm{Tr}(\mathrm{Cov}(J_{ii}^{\top}\mathbf{r}_{ii}, J_{ii}^{\top}\mathbf{r}_{-ii})) \\ &\geq \|\Sigma_{ii}\|_1 + \|\Sigma_{-ii}\|_1 - 2\sqrt{\|\Sigma_{ii}\|_1\|\Sigma_{-ii}\|_1} \\ &> \|\Sigma_{ii}\|_1.\end{aligned}$$

The first inequality uses the Cauchy-Schwarz inequality for covariance, with equality if and only if the true AR gradient $-J_{ii}^{\top}\mathbf{r}_{ii}$ and the CR contamination $-J_{ii}^{\top}\mathbf{r}_{-ii}$ are perfectly negatively correlated. The second inequality directly follows from assumption 1. Hence, the mixed AR gradient is strictly less stable than the true AR gradient.

Alternating Training Strategy To mitigate gradient entanglement, we separately optimize the AR and CR paths. Specifically, we repeat a two-step alternating optimization (AO) cycle until convergence.

- Step 1. $\theta_{\text{AR}}^{(i+1)} \leftarrow \arg\min_{\theta_{\text{AR}}} \|\mathbf{Y}_{t+1} - (\mathbf{F}_{\text{AR}} + \mathbf{F}_{\text{CR}}^{(i)}) * \mathbf{X}_t\|_2^2 + R_{\text{AR}}(\theta_{AR})$.
- Step 2. $\theta_{\text{CR}}^{(i+1)} \leftarrow \arg\min_{\theta_{\text{CR}}} \|\mathbf{Y}_{t+1} - (\mathbf{F}_{\text{AR}}^{(i+1)} + \mathbf{F}_{\text{CR}}) * \mathbf{X}_t\|_2^2 + R_{\text{CR}}(\theta_{CR})$.

$R_{\text{AR}}(\cdot)$ and $R_{\text{CR}}(\cdot)$ are regularizers for the AR module and the CR module, respectively. In practice, we initialize two independent AMSGrad optimizers (Reddi et al., 2018). For each batch, we first apply the AR regularizer and optimize the AR parameters with the CR parameters frozen; we then alternate the regularizer and freeze the AR parameters to optimize the CR parameters on the same batch. Both subproblems are run for a small but non-trivial number of inner iterations. We use AMSGrad due to its convergence guarantee, aligning with alternating optimization algorithms that require sufficient descent and convergence of each subproblem (Razaviyayn et al., 2013; Bolte et al., 2014). For the same reason, we always update the AR path prior to the CR path.

The proposed alternating training strategy isolates the two parameter blocks. Intuitively, each subproblem is easier to optimize than the original joint problem due to reduced dimensionality. It also enables distinct learning schedules tailored to AR and CR, which may intrinsically require different step sizes and regularization. Regarding gradient stability, by the law of total variance, the gradient covariance under joint training for θ_{ii} is

$$\text{Cov}(J_{ii}^\top \mathbf{r}_i) = \mathbb{E}_{\theta_{\text{CR}}}(\text{Cov}(J_{ii}^\top \mathbf{r}_i \mid \theta_{\text{CR}})) + \text{Cov}_{\theta_{\text{CR}}}(\mathbb{E}(J_{ii}^\top \mathbf{r}_i \mid \theta_{\text{CR}})). \tag{12}$$

In Step 1 of alternating training, θ_{CR} is held fixed, so the gradient covariance for θ_{ii} is $\text{Cov}(J_{ii}^\top \mathbf{r}_i \mid \theta_{\text{CR}})$. Averaging over the distribution of θ_{CR} yields

$$\mathbb{E}_{\theta_{\text{CR}}}(\text{Cov}(J_{ii}^\top \mathbf{r}_i \mid \theta_{\text{CR}})), \tag{13}$$

which is precisely the first term in equation 12. The between-path source of gradient variability is removed. This particularly benefits the AR path, as the CR path usually has noticeably more parameters and slower convergence, making the second term in equation 12 non-negligible.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on seven standard LTSF benchmarks: **Weather**, **Traffic**, **Electricity**, and the ETT family (**ETTh1**, **ETTh2**, **ETTm1**, **ETTm2**). Each dataset is chronologically split into train/validation/test sets, and we forecast horizons of $\{96, 192, 336, 720\}$ steps. The input length is fixed to 512, following the convention of PatchTST (Nie et al., 2023). This setting provides a sufficiently long receptive field to capture both autoregressive stability and cross-variable interactions.

Baselines. We benchmark against representative methods spanning different modeling paradigms: *RLinear* (Li et al., 2023) and *DLinear* (Zeng et al., 2023) (channel-independent/pooled linear models), *PatchTST* (Nie et al., 2023) (patch-based CI/CM Transformer), *iTransformer* (Liu et al., 2023) (variable-token attention for cross-variable modeling), *Informer* (Zhou et al., 2021) (sparse attention), and *TimeBase* (Huang et al., 2025) (autoregression-centric backbone). These baselines cover both autoregressive and cross-variable approaches, providing a comprehensive set of alternative state-of-the-art approaches to evaluate our framework.

Implementation. In our dual-path framework, we instantiate the **AR path with RLinear** (Li et al., 2023) and the **CR path with iTransformer** (Liu et al., 2023), aiming to highlight the effect of *alternating optimization* rather than architectural refinements. RLinear cleanly models channel-wise autoregression, while iTransformer captures cross-variable dependencies via variable-level self-attention. Both paths use RevIN normalization and are additively combined after denormalization. We adapt Alternating Optimization (AO)

Model		Ours		TimeBase		iTransformer		RLinear		PatchTST		DLinear		Informer	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.147**	**0.200**	0.151	0.204	0.168	0.218	0.171	0.223	0.150	0.205	0.170	0.229	0.350	0.410
	192	**0.187**	**0.240**	0.192	0.241	0.210	0.255	0.216	0.260	0.194	0.242	0.215	0.275	0.420	0.430
	336	**0.238**	0.281	0.244	0.282	0.260	0.291	0.261	0.294	0.242	**0.279**	0.258	0.309	0.580	0.549
	720	**0.313**	0.338	0.317	0.336	0.331	0.341	0.323	0.339	0.314	**0.332**	0.319	0.359	0.920	0.699
Traffic	96	**0.357**	0.256	0.392	0.259	0.367	0.272	0.395	0.272	0.360	**0.249**	0.394	0.274	0.739	0.412
	192	**0.377**	0.267	0.413	0.274	0.382	0.269	0.407	0.276	0.379	**0.256**	0.406	0.279	0.777	0.435
	336	**0.390**	0.276	0.427	0.287	0.395	0.279	0.416	0.282	0.392	**0.264**	0.415	0.285	0.775	0.450
	720	0.443	0.306	0.466	0.301	**0.418**	0.289	0.454	0.302	0.432	**0.286**	0.453	0.307	0.820	0.460
Electricity	96	0.132	0.229	0.136	0.229	0.132	0.228	0.139	0.244	**0.129**	**0.222**	0.135	0.232	0.300	0.399
	192	0.149	0.244	0.159	0.255	0.153	0.248	0.150	0.242	**0.147**	**0.240**	0.148	0.245	0.327	0.418
	336	0.166	0.262	0.172	0.288	0.168	0.265	0.166	0.260	**0.163**	**0.259**	0.164	0.265	0.334	0.432
	720	0.206	0.295	0.219	0.301	**0.194**	**0.286**	0.212	0.300	0.197	0.290	0.198	0.295	0.356	0.429
ETTh1	96	**0.364**	0.395	0.384	0.392	0.399	0.425	0.367	**0.392**	0.370	0.400	0.378	0.404	0.932	0.766
	192	**0.399**	**0.417**	0.432	0.462	0.426	0.442	0.402	0.422	0.412	0.428	0.405	0.417	1.001	0.780
	336	**0.412**	**0.430**	0.445	0.473	0.457	0.464	0.419	0.617	0.421	0.439	0.452	0.457	1.030	0.780
	720	**0.433**	**0.453**	0.449	0.482	0.630	0.574	0.451	0.463	0.447	0.467	0.502	0.513	1.139	0.852
ETTh2	96	**0.272**	**0.337**	0.401	0.439	0.297	0.356	0.275	0.339	0.279	0.341	0.282	0.348	1.542	0.957
	192	**0.332**	0.382	0.451	0.467	0.377	0.405	0.336	**0.371**	0.341	0.381	0.359	0.403	3.791	1.522
	336	0.330	0.390	0.458	0.482	0.424	0.440	**0.324**	0.385	0.328	**0.383**	0.440	0.454	4.200	1.640
	720	**0.374**	**0.418**	0.502	0.510	0.438	0.461	0.415	0.445	0.379	0.422	0.608	0.560	3.660	1.620
ETTm1	96	**0.292**	**0.338**	0.314	0.356	0.311	0.365	0.310	0.351	0.294	0.346	0.304	0.348	0.626	0.549
	192	**0.331**	**0.360**	0.339	0.370	0.348	0.385	0.338	0.367	0.333	0.370	0.336	0.367	0.725	0.621
	336	**0.365**	**0.380**	0.374	0.392	0.379	0.405	0.369	0.385	0.370	0.392	0.374	0.395	1.002	0.745
	720	0.426	**0.414**	0.424	0.425	0.443	0.444	0.429	0.415	**0.415**	0.419	0.427	0.425	1.139	0.841
ETTm2	96	**0.160**	**0.249**	0.167	0.257	0.178	0.272	0.163	0.251	0.166	0.256	0.165	0.256	0.352	0.467
	192	0.219	**0.290**	0.221	0.293	0.241	0.315	**0.217**	0.292	0.223	0.296	0.226	0.306	0.599	0.579
	336	**0.266**	**0.323**	0.273	0.327	0.290	0.344	0.271	0.326	0.273	0.329	0.274	0.335	1.277	0.882
	720	**0.352**	**0.382**	0.368	0.389	0.376	0.397	**0.360**	0.387	0.361	0.385	0.380	0.408	2.892	1.219

Table 1: **Full Results.** Comparison across seven datasets (Weather, Traffic, Electricity, ETTh1, ETTh2, ETTm1, ETTm2) and four prediction horizons (96/192/336/720). Columns are ordered as *Ours*, TimeBase, iTransformer, RLinear, PatchTST, DLinear, and Informer. Metrics are MSE and MAE (lower is better). **Bold** numbers denote the best (lowest) performance in each row, while underlined numbers denote the second best.

in training: AR parameters are updated with CR frozen and vice versa on each mini-batch, each subproblem optimized by AMSGrad with early stopping. We scale $R_{\text{AR}}(\cdot)$ and $R_{\text{CR}}(\cdot)$ by prediction length to account for the sensitivity of the ℓ_1 regularizer to parameter count.

Metrics. We report Mean Squared Error (MSE) and Mean Absolute Error (MAE) averaged across all variates, following LTSF conventions. Lower values indicate better predictive performance.

4.2 Main Results

Table 1 reports the results across all datasets and horizons. Our method achieves state-of-the-art or second-best performance in the vast majority of cases. In *ETT family* and *Weather*, our model substantially outperforms baselines, demonstrating the benefit of stabilizing autoregression while still modeling intermittent cross-variable interactions. For *Electricity*, our results remain competitive with PatchTST and RLinear, confirming that decoupling does not compromise performance in CI-friendly regimes. On *Traffic*, we outperform baselines on short and mid horizons but are slightly behind iTransformer at 720 steps. Taken together, these results indicate that decoupling AR and CR, when paired with alternating optimization, consistently improves predictive stability and accuracy across diverse datasets, and the gains are most visible at longer horizons, where joint training tends to degrade.

4.3 Ablation Studies

We conduct ablation studies to better understand the contributions of the two key components in our framework: the decoupled AR and CR paths, and the alternating optimization strategy. The first set of experiments disentangle the effect of each path by comparing against their standalone counterparts (RLinear and iTrans-

Horizon (steps)	ETTh1		ETTh2		ETTm1		ETTm2	
	w/ AO	w/o AO	w/ AO	w/o AO	w/ AO	w/o AO	w/ AO	w/o AO
96	**0.364/0.395**	0.437/0.453	**0.272/0.337**	0.321/0.381	**0.292/0.338**	0.316/0.364	**0.160/0.249**	0.175/0.263
192	**0.399/0.417**	0.471/0.473	**0.332/0.382**	0.371/0.414	**0.331/0.360**	0.359/0.387	**0.219/0.290**	0.243/0.313
336	**0.412/0.430**	0.472/0.479	**0.330/0.390**	0.374/0.426	**0.365/0.380**	0.392/0.406	**0.266/0.323**	0.289/0.341
720	**0.433/0.453**	0.510/0.506	**0.374/0.418**	0.440/0.463	**0.426/0.414**	0.449/0.433	**0.352/0.382**	0.381/0.401

(a) ETTh1, ETTh2, ETTm1, ETTm2

Horizon (steps)	Electricity		Traffic		Weather	
	w/ AO	w/o AO	w/ AO	w/o AO	w/ AO	w/o AO
96	**0.132/0.229**	0.137/0.233	**0.357/0.256**	0.364/0.258	**0.147/0.200**	0.157/0.211
192	**0.149/0.244**	0.161/0.256	**0.377/0.267**	0.386/0.269	**0.187/0.240**	0.203/0.253
336	**0.166/0.262**	0.179/0.274	**0.390**/0.276	0.394/**0.275**	**0.238/0.281**	0.249/0.288
720	0.206/0.295	**0.198/0.291**	**0.443**/0.306	0.481/**0.301**	**0.313/0.338**	0.321/0.339

(b) Electricity, Traffic, Weather

Table 2: Ablation study of our method with and without Alternating Optimization (AO) optimization. Each cell shows MSE/MAE, and **bold** indicates the better (lower) value for each metric.

former), while the second set isolates the role of alternating optimization (AO) by contrasting alternating training with joint optimization. Together, these analyses clarify how architectural decoupling and optimization strategy jointly contribute to the performance of ALTTS.

AR vs. CR Path Contributions. To further examine the dual-path design, we contrast the performance of RLinear (AR-only) and iTransformer (CR-only) from Table 1 with our ALTTS model. RLinear excels on Electricity due to strong periodic autoregression but underperforms on Weather and Traffic where cross-variable relations matter. iTransformer, conversely, benefits datasets with strong cross-relations but is less effective on ETT where autoregression dominates. Our decoupled ALTTS model with AR and CR paths, trained with alternating optimization, consistently outperforms both standalone variants, demonstrating that the two paths are complementary and that alternating optimization is crucial for integrating them effectively.

Effect of Alternating Optimization. Table 2 compares our model with and without AO; in the latter case, the AR and CR modules are optimized jointly using a single optimizer. Across all benchmarks, AO reduces both MSE and MAE, with the improvement being most pronounced at long horizons (e.g., ETTh1 at 720: $0.433/0.453$ vs. $0.510/0.506$). This validates our theoretical analysis that alternating optimization mitigates gradient entanglement between AR and CR, leading to more stable convergence.

5 CONCLUSIONS

In this paper, we have introduced ALTTS, a dual-path framework for multivariate time series forecasting that *decouples* autoregression (AR) and cross-relation (CR) modeling and coordinates them via *alternating optimization*. Grounded in an analysis of gradient entanglement under joint AR–CR training, we instantiated the framework with RLinear for AR and iTransformer for CR to emphasize optimization over architectural novelty. Across seven multivariate LTSF benchmarks and four horizons, ALTTS delivers competitive or superior accuracy, with the largest gains at long horizons where joint training is most unstable. Ablations confirm that AR/CR separation is complementary and that alternating optimization is the key driver of stability and performance. In summary, ALTTS is the first deep learning framework to explicitly decouple autoregression and cross-variable dependency via alternating optimization, supported by a theoretical analysis of gradient entanglement. Extensive experiments on seven benchmarks show consistent improvements over strong linear, Transformer-based, and hybrid baselines. Finally, we highlight training schedules as a design variable, suggesting a broader paradigm where optimization principles inform neural network architecture.

Reproducibility Statement

All datasets used in this paper are publicly available, and we follow the established preprocessing and evaluation protocol of Nie et al. (2023). We release complete code, training scripts, and instructions to reproduce all reported results. Code is available at `https://anonymous.4open.science/r/AltTS-Official-Implementation-F379/README.md`.

References

Chirag Agarwal, Daniel D'souza, and Sara Hooker. Estimating example difficulty using variance of gradients. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10358–10368, 2022. doi: 10.1109/CVPR52688.2022.01012.

Hassan Akbari, Dan Kondratyuk, Yin Cui, Rachel Hornung, Huisheng Wang, and Hartwig Adam. Alternating gradient descent and mixture-of-experts for integrated multimodal perception. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145. PMLR, 06–11 Aug 2017.

Jushan Bai and Serena Ng. Determining the number of factors in approximate factor models. *Econometrica*, 70(1):191–221, 2002. ISSN 00129682, 14680262.

Sumanta Basu and George Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535 – 1567, 2015. doi: 10.1214/15-AOS1315.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.*, 146(1–2):459–494, August 2014. ISSN 0025-5610. doi: 10.1007/s10107-013-0701-9.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010. ISSN 1935-8237. doi: 10.1561/2200000016.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2024.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1675–1685. PMLR, 09–15 Jun 2019.

Robert F. Engle and C. W. J. Granger. Co-integration and error correction: Representation, estimation, and testing. *Econometrica*, 55(2):251–276, 1987. ISSN 00129682, 14680262.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

Qihe Huang, Zhengyang Zhou, Kuo Yang, Zhongchao Yi, Xu Wang, and Yang Wang. Timebase: The power of minimalism in efficient long-term time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *ICLR (Poster)*, 2015.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2017.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping, 2023.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. *arXiv preprint arXiv:2405.00946*, 2024.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Songbo Wang, and Yongxiang Wang. Petformer: Long-term time series forecasting via placeholder-enhanced transformer. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 9(2):1189–1201, 2025. doi: 10.1109/TETCI.2024.3502437.

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 Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

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

Donghao Luo and Xue Wang. Deformabletst: Transformer for time series forecasting without over-reliance on patching. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 88003–88044. Curran Associates, Inc., 2024.

Zhengxiong Luo, Yan Huang, Shang Li, Liang Wang, and Tieniu Tan. Unfolding the alternating optimization for blind super resolution. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.

Whitney K. Newey and Kenneth D. West. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, 55(3):703–708, 1987. ISSN 00129682, 14680262.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

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

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *SIGKDD*, pp. 1185–1196, 2025.

Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013. doi: 10.1137/120891009.

Meisam Razaviyayn, Mingyi Hong, Zhi-Quan Luo, and Jong-Shi Pang. Parallel successive convex approximation for nonsmooth nonconvex optimization. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations*, 2021.

Christopher A. Sims. Macroeconomics and reality. *Econometrica*, 48(1):1–48, 1980. ISSN 00129682, 14680262.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and JUN ZHOU. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, Jun. 2023.

Jinshan Zeng, Tim Tsz-Kit Lau, Shaobo Lin, and Yuan Yao. Global convergence of block coordinate descent in deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7313–7323. PMLR, 09–15 Jun 2019.

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

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 *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, volume 35, pp. 11106–11115. AAAI Press, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022.

A Appendix

A.1 LLM Usage Disclosure

In accordance with the ICLR 2026 policy on large language models (LLMs), we disclose the use of ChatGPT as a writing assist tool. Specifically, the LLM was employed to polish the presentation of text, improve clarity, and refine phrasing. The authors carefully reviewed and edited all LLM-assisted text to ensure accuracy and alignment with the intended scientific contributions. No part of the ideation, or substantive analysis relied on the LLM.

A.2 Dataset Details

Dataset	Variates	Freq.	Input Len.	Pred. Len.	Domain
ETTh1	7	Hourly	512	96–720	Electricity
ETTh2	7	Hourly	512	96–720	Electricity
ETTm1	7	15 min	512	96–720	Electricity
ETTm2	7	15 min	512	96–720	Electricity
Weather	21	10 min	512	96–720	Weather
Electricity	321	Hourly	512	96–720	Electricity
Traffic	862	Hourly	512	96–720	Transport

Table 3: **Dataset statistics.**

We evaluate long-term forecasting performance on seven widely used multivariate benchmarks: **Weather**, **Traffic**, **Electricity**, and the ETT family (**ETTh1**, **ETTh2**, **ETTm1**, **ETTm2**). These datasets are standard in the LTSF community and cover diverse domains including weather monitoring, traffic flow, and energy consumption. Following the established protocol of Nie et al. (2023); Wu et al. (2021), we split the data chronologically into training/validation/test sets with a ratio of 6:2:2 for the ETT datasets and 7:1:2 for the others. The input length is fixed to 512 across all experiments, and prediction horizons are $\{96, 192, 336, 720\}$. Key statistics are summarized in Table 3.

Weather. Records 21 meteorological indicators (e.g., temperature, humidity, wind speed) every 10 minutes throughout 2020 in Germany.

Traffic. Hourly road occupancy rates measured by 862 sensors on San Francisco Bay Area freeways between 2015 and 2016.

Electricity. Hourly electricity consumption (kWh) of 321 customers from 2012 to 2014.

ETT. The Electricity Transformer Temperature datasets include two hourly datasets (ETTh1, ETTh2) and two 15-minute datasets (ETTm1, ETTm2). Each contains seven oil temperature and load features collected from electricity transformers between July 2016 and July 2018.