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 forecasting requires learning both within-variable temporal dependencies and cross-variable interactions. When these components are optimized jointly, their predictions enter a shared residual, coupling their learning signals. We introduce ALTTS, a dual-path forecasting framework that explicitly separates temporal and cross-variable modeling and coordinates their adaptation through staged alternating optimization. The framework combines a linear temporal predictor with a masked cross-variable attention module. Its training procedure first establishes a temporal predictor, fits a cross-variable residual correction, and then alternates between the branches with block relaxation. Our analysis characterizes residual-induced gradient coupling, and local learning-signal diagnostics illustrate how this interaction can affect temporal updates. Experiments on seven forecasting benchmarks demonstrate competitive accuracy using standard architectural components. These results highlight branch coordination as an effective design dimension for multivariate forecasting. Code available at: `anonymous github`.

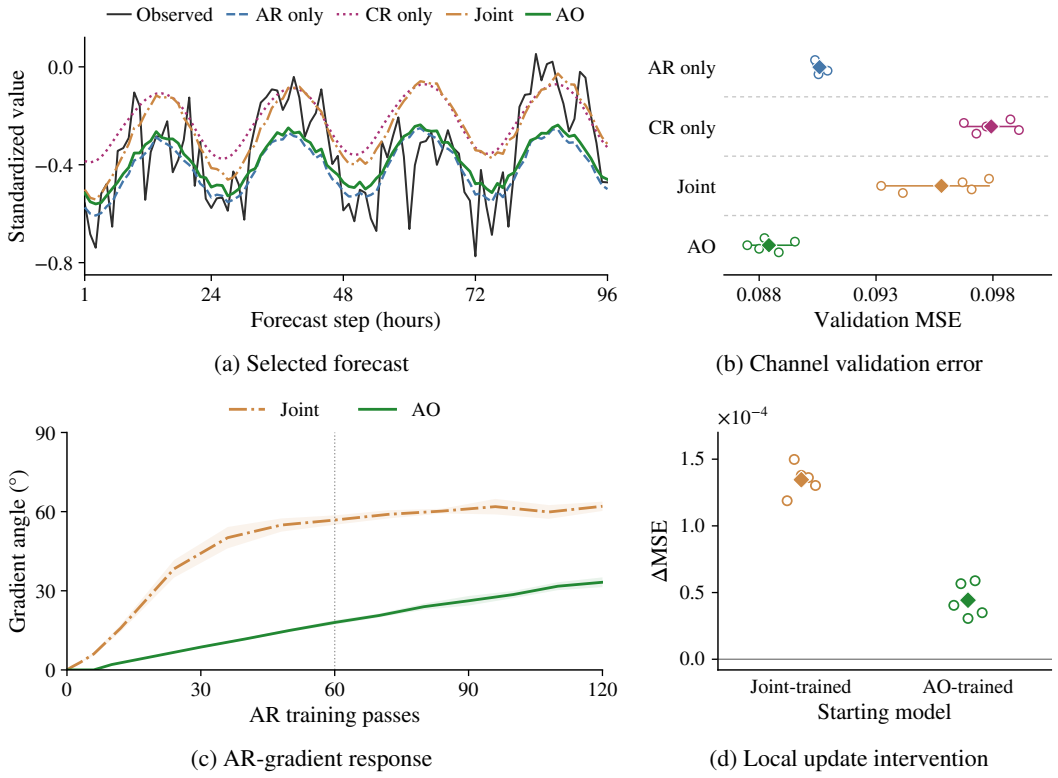
1 INTRODUCTION

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). In those statistical methods, modeling cross-variable dependencies is fundamental to understanding the multivariate system.

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; 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 for effective forecasting.

Architectural separation alone does not separate the learning signals of temporal and cross-variable predictors. Under a combined forecasting loss, both predictions enter the residual used to update either branch. Cross-variable modeling therefore affects not only the forecast but also how temporal dependencies are learned. Figure 1 illustrates a selected case where joint training underperforms the temporal predictor alone, whereas AO improves over both standalone branches and joint training. The accompanying diagnostics show that the CR contribution can alter AR gradients and adversely

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Selected forecast

(b) Channel validation error

(c) AR-gradient response

(d) Local update intervention

Figure 1: **An illustrative forecasting case and local learning-signal diagnostics.** (a) Forecasts from AR only, CR only, joint training, and AO. (b) Validation MSE for the selected channel. (c) AR-gradient angle with versus without the CR contribution in the learning residual. (d) Validation MSE after the ordinary update minus that after the modified update; positive values favor the modified update. Examples are taken from ETTh dataset with input length 512 and prediction horizon 96.

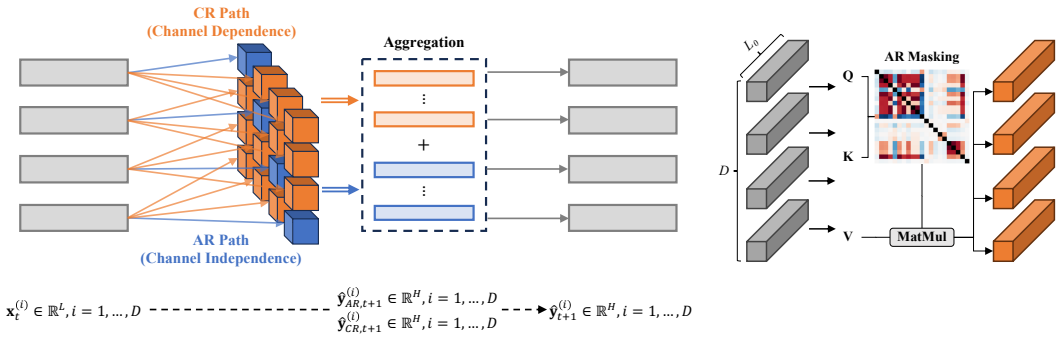
affect local updates, with smaller gradient angles under AO indicating less directional perturbation. These observations motivate explicit coordination of the two branches during optimization.

We propose ALTTS, a dual-path forecasting framework that separates temporal and cross-variable prediction and coordinates their learning through staged alternating optimization. The training procedure establishes compatible initial predictors before alternating between the branches. Block relaxation then controls how much of each proposed update is committed to the combined model. Using standard linear and attention-based modules, ALTTS provides a simple implementation of this optimization-driven design. Our contributions are threefold:

1. We characterize gradient coupling between temporal and cross-variable predictors through their shared forecasting residual and examine its local effects on temporal learning.
2. We develop ALTTS, combining a dual-path architecture with staged initialization and block-relaxed alternating optimization to coordinate branch adaptation.
3. We evaluate ALTTS on seven multivariate forecasting benchmarks, demonstrating competitive accuracy and examining the role of its modeling and training choices.

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 adapt the Transformer (Vaswani et al., 2017) 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,



(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 a diagonal mask excludes direct self-value attention.

2024). Further investigations show that simple models, such as linear layers (Zeng et al., 2023; Li et al., 2023), can achieve comparable or superior performance to Transformers. More recently, OLinear (Yue et al., 2025) demonstrates that linear forecasters can remain highly competitive by operating in an orthogonally transformed domain to alleviate entangled temporal dependencies.

Dependency Modeling Dependency modeling distinguishes predictors that use only each variable's history from those that also use multivariate context. Channel-independent (CI) predictors may share parameters across variables while keeping their input histories separate. PatchTST (Nie et al., 2023) combines this design with temporal patching. Channel-dependent (CD) methods explicitly model cross-variable structure, which can provide additional predictive information when other variables contain signals complementary to the target's history. Crossformer (Zhang & Yan, 2023) uses two-stage attention to capture cross-time and cross-dimension dependencies. iTransformer (Liu et al., 2023) tokenizes variables and applies attention to their temporal embeddings, implicitly representing multivariate correlations. To address high channel dimensionality, channel clustering (CC), analogous to temporal patching, groups heterogeneous variables to preserve instantaneous correlations, as in DUET (Qiu et al., 2025) and TimeFilter (Hu et al., 2025). The two-stage temporal-spatial paradigm is widely adopted among CD methods. Although recent studies also explore structures compatible with both CI and CD settings (Lin et al., 2025), further research on integrating autoregressive and cross-dimension modeling remains necessary.

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. We apply alternating parameter-block updates to coordinate temporal and cross-variable prediction in LTSF. Our finite-Adam implementation uses block relaxation to control the committed parameter displacement; classical convergence guarantees for exact or sufficiently decreasing block updates are not assumed.

3 METHODOLOGY

3.1 GENERAL STRUCTURE

In multivariate time series forecasting, let $\mathbf{X}_t = [\mathbf{x}_t^{(1)}, \ldots, \mathbf{x}_t^{(D)}]^\top \in \mathbb{R}^{D \times L}$ denote the lookback window at step t, where $\mathbf{x}_t^{(i)} = [x_{t-L+1}^{(i)}, \ldots, x_t^{(i)}]^\top \in \mathbb{R}^L$ contains the historical observations of

variable i. Here, D is the number of variables and L is the lookback length. The forecasting target is $\mathbf{Y}_{t+1} = [\mathbf{y}_{t+1}^{(1)}, \ldots, \mathbf{y}_{t+1}^{(D)}]^\top \in \mathbb{R}^{D \times H}$, where $\mathbf{y}_{t+1}^{(i)} = [x_{t+1}^{(i)}, \ldots, x_{t+H}^{(i)}]^\top \in \mathbb{R}^H$ contains the next H observations. For brevity, we write $\mathbf{x}_i := \mathbf{x}_t^{(i)}$ and $\mathbf{y}_i := \mathbf{y}_{t+1}^{(i)}$ below.

We parameterize each predicted series as

$$\hat{\mathbf{y}}_i = A_i(\mathbf{x}_i; \theta_A) + C_i(\mathbf{X}_t; \theta_C), \qquad i = 1, \ldots, D. \tag{1}$$

The temporal branch $A_i : \mathbb{R}^L \to \mathbb{R}^H$ uses the complete history of variable i, while the multivariate correction $C_i : \mathbb{R}^{D \times L} \to \mathbb{R}^H$ conditions on the historical window across variables. The branches have disjoint trainable parameter blocks θ_A and θ_C. Autoregressive modeling here refers to prediction from a variable's own history, with the full forecast horizon generated directly.

3.2 AR-CR Structural Decoupling

Temporal and cross-variable predictors capture different sources of forecasting information, but their learning signals interact through the combined prediction residual. Figure 1(c,d) shows how the cross-variable contribution changes the temporal gradient and can adversely affect a local update. We address this interaction through separate parameterization and coordinated branch adaptation.

We instantiate Equation 1 with a linear temporal predictor and a masked cross-variable attention module.

Shared normalization. We apply parameter-free RevIN to each input window. For variable i, the historical mean and scale are

$$\mu_i = \frac{1}{L} \sum_{\ell=1}^{L} X_{i\ell}, \qquad s_i = \sqrt{\frac{1}{L} \sum_{\ell=1}^{L} (X_{i\ell} - \mu_i)^2 + \epsilon}, \tag{2}$$

where $\epsilon = 10^{-5}$. Both branches receive the same normalized inputs $\tilde{X}_{i\ell} = (X_{i\ell} - \mu_i)/s_i$. The statistics are computed over the lookback dimension and broadcast over the prediction horizon.

Auto-Regression Path. Following RLinear (Li et al., 2023), we use a linear mapping from each normalized variable history to its forecast:

$$a_i(\tilde{\mathbf{x}}_i) = W_{A,i} \tilde{\mathbf{x}}_i + b_{A,i}. \tag{3}$$

Each prediction uses only the corresponding variable's history. Parameters may be variable-specific or shared across variables, depending on the configuration. The AR branch uses ℓ_1 regularization.

Cross-Relation Path. We use an inverted Transformer encoder to model interactions between variable tokens. A learned affine projection followed by ReLU maps each normalized history to an L_0-dimensional embedding:

$$Z_0 = \mathrm{Embedding}(\tilde{\mathbf{X}}_t) \in \mathbb{R}^{D \times L_0}. \tag{4}$$

For each of n_{head} attention heads, learned projections produce $Q, K, V \in \mathbb{R}^{D \times d_h}$, where $d_h = L_0 / n_{\mathrm{head}}$. Cross-Relation Self-Attention is

$$\mathrm{CRSA}(Z_0) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} + M \right) V, \tag{5}$$

with $M_{ii} = -\infty$ and $M_{ij} = 0$ for $i \neq j$. The mask excludes direct self-value attention. Queries and residual connections retain the target variable's context, while attention aggregates information from other variables. The interaction weights are recomputed for each input window.

We denote by MHSA the concatenation and output projection of the masked CRSA heads defined above, yielding an output in $\mathbb{R}^{D \times L_0}$. Each encoder block applies

$$\begin{aligned} Z_1 &= \mathrm{LayerNorm}\left(Z_0 + \mathrm{Dropout}(\mathrm{MHSA}(Z_0))\right), \\ Z_2 &= \mathrm{LayerNorm}\left(Z_1 + \mathrm{Dropout}(\mathrm{MLP}(Z_1))\right), \end{aligned} \tag{6}$$

where the MLP is a two-layer feedforward network. A linear prediction head maps each final variable token to a length-H normalized correction $c_i(\tilde{\mathbf{X}}_t)$.

Prediction aggregation. The normalized branch outputs are added before a single denormalization:

$$\hat{\mathbf{y}}_i = \mu_i \mathbf{1}_H + s_i \left[a_i(\tilde{\mathbf{x}}_i) + c_i(\tilde{\mathbf{X}}_t) \right]. \tag{7}$$

Accordingly, the contributions in Equation 1 are $A_i = \mu_i \mathbf{1}_H + s_i a_i$ and $C_i = s_i c_i$.

3.3 Block Alternating Optimization

Residual-induced gradient coupling. Let $\widehat{Y} = A_{\theta_A}(X) + C_{\theta_C}(X)$, where A and C denote the branch contributions after accounting for the shared normalization and denormalization. Their trainable parameter blocks are disjoint. For the following analysis, predictions and targets are vectorized in $\mathbb{R}^{DH}$. Expressions are written for one input window; minibatch gradients average the corresponding expressions.

The regularized mean-squared-error objective is

$$\mathcal{L} = \frac{1}{DH} \| A_{\theta_A}(X) + C_{\theta_C}(X) - Y \|_2^2 + R_A(\theta_A) + R_C(\theta_C). \tag{8}$$

Let $J_A = \partial A_{\theta_A}(X)/\partial \theta_A$ and let $r_A \in \partial R_A(\theta_A)$ denote the regularizer subgradient, using the same selection in both expressions below. The AR update signal is

$$g_A = \frac{2}{DH} J_A^\top \big(A_{\theta_A}(X) + C_{\theta_C}(X) - Y \big) + r_A. \tag{9}$$

Omitting the CR contribution from the learning residual gives the reference signal

$$g_A^{(0)} = \frac{2}{DH} J_A^\top \big(A_{\theta_A}(X) - Y \big) + r_A. \tag{10}$$

Their difference is

$$g_A - g_A^{(0)} = \frac{2}{DH} J_A^\top C_{\theta_C}(X). \tag{11}$$

Thus, the CR prediction influences temporal learning despite the separation of trainable parameters. We refer to this shared-residual interaction as gradient coupling.

Effect on gradient variability. At fixed parameters, let g_A and $g_A^{(0)}$ denote the corresponding gradients averaged over a random minibatch. Write $u = g_A^{(0)}$ and $v = g_A - g_A^{(0)}$, and assume finite second moments. Let $\Sigma_u = \mathrm{Cov}(u)$, $\Sigma_v = \mathrm{Cov}(v)$, and $\Gamma_{uv} = \mathrm{Cov}(u, v)$. Then

$$\mathrm{Cov}(g_A) = \Sigma_u + \Sigma_v + \Gamma_{uv} + \Gamma_{uv}^\top. \tag{12}$$

By the Cauchy–Schwarz inequality,

$$\mathrm{tr}\, \mathrm{Cov}(g_A) \geq \mathrm{tr}\, \Sigma_u + \mathrm{tr}\, \Sigma_v - 2\sqrt{\mathrm{tr}\, \Sigma_u \ \mathrm{tr}\, \Sigma_v}. \tag{13}$$

In particular, $\mathrm{tr}\, \Sigma_v > 4\, \mathrm{tr}\, \Sigma_u$ is sufficient for the coupled gradient to have greater variance than the reference. This condition characterizes a possible increase in gradient variability. Its predictive consequences also depend on the update direction. We therefore complement the analysis with paired local interventions that evaluate the resulting change in validation error.

In Figure 1(d), paired AR updates start from the same checkpoint and have matched parameter-displacement norms. The modified update omits CR from the learning residual, while both evaluated forecasts retain the CR prediction. Panels (a,b) show a selected channel, and panels (c,d) aggregate all channels. The illustration uses ordinary AO.

Staged initialization. We first fit AR to the forecasting target. With the selected AR parameters fixed, we then fit CR to the residual $Y - A_{\theta_A}(X)$. The CR output-head weights and biases are initialized to zero, so its initial contribution is zero. Each stage retains the checkpoint with the lowest validation MSE, including the checkpoint before its first update. The resulting parameters initialize the subsequent block alternating optimization.

Algorithm 1 Staged Alternating Optimization with Block Relaxation

Require: Training data $\mathcal{D}_{\mathrm{tr}}$, validation data $\mathcal{D}_{\mathrm{val}}$; block lengths N_A, N_C; relaxation coefficients α_A, α_C; number of alternating cycles K
Ensure: Validation-selected parameters $(\theta_A^\star, \theta_C^\star)$

$\theta_A \leftarrow \textsc{FitAR}(\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{val}})$
$\theta_C \leftarrow \textsc{FitResidualCR}(\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{val}}, \theta_A)$
Initialize separate Adam states s_A, s_C
$(\theta_A^\star, \theta_C^\star) \leftarrow (\theta_A, \theta_C)$
$v^\star \leftarrow \textsc{ValMSE}(\theta_A, \theta_C, \mathcal{D}_{\mathrm{val}})$
for $k = 1, \ldots, K$ **do**
    $(\widetilde{\theta}_A, s_A) \leftarrow \textsc{ARBlock}(\theta_A, \theta_C, s_A, N_A)$ ▷ CR fixed
    $\theta_A \leftarrow \theta_A + \alpha_A(\widetilde{\theta}_A - \theta_A)$
    if $N_C > 0$ **then**
        $(\widetilde{\theta}_C, s_C) \leftarrow \textsc{CRBlock}(\theta_A, \theta_C, s_C, N_C)$ ▷ AR fixed
        $\theta_C \leftarrow \theta_C + \alpha_C(\widetilde{\theta}_C - \theta_C)$
    end if
    $v \leftarrow \textsc{ValMSE}(\theta_A, \theta_C, \mathcal{D}_{\mathrm{val}})$
    if $v < v^\star$ **then**
        $v^\star \leftarrow v$
        $(\theta_A^\star, \theta_C^\star) \leftarrow (\theta_A, \theta_C)$
    end if
end for
return $(\theta_A^\star, \theta_C^\star)$

Block-relaxed alternating updates. Let $N_A \geq 1$ and $N_C \geq 0$ be integer block lengths measured in full training passes, and let $0 < \alpha_A, \alpha_C \leq 1$ be relaxation coefficients. Starting from (θ_A^k, θ_C^k), we optimize AR for N_A passes with the CR parameters fixed, producing a proposal $\widetilde{\theta}_A^{k+1}$. The committed parameters are

$$\theta_A^{k+1} = \theta_A^k + \alpha_A(\widetilde{\theta}_A^{k+1} - \theta_A^k). \tag{14}$$

We then optimize CR for N_C passes with the committed AR parameters fixed, obtaining $\widetilde{\theta}_C^{k+1}$, and set

$$\theta_C^{k+1} = \theta_C^k + \alpha_C(\widetilde{\theta}_C^{k+1} - \theta_C^k). \tag{15}$$

Setting $\alpha_A = \alpha_C = 1$ gives ordinary alternating updates from the same initialization; smaller positive coefficients reduce the committed block displacement.

Each proposal is computed using a finite sequence of Adam updates. The learning rate controls individual inner updates, whereas relaxation controls the displacement of the complete block proposal. We initialize separate optimizer states at the continuation boundary and retain them across subsequent blocks, including after interpolation.

At an identical model state, minibatch, and stochastic realization, freezing the other branch does not change the active partial derivative. The method therefore acts by controlling the optimization trajectory rather than algebraically removing the shared-residual term.

Training procedure. Algorithm 1 summarizes training for a fixed configuration. Each alternating cycle performs N_A AR passes and N_C CR passes using Adam and the corresponding regularized objective, giving KN_A additional AR passes over K cycles. Freezing a branch holds its parameters and optimizer state fixed; dropout remains active during training. Adam states persist across blocks and are not interpolated when parameter proposals are relaxed.

Table 1: **Full Results.** Forecasting performance on seven datasets with input length 512 and prediction horizons 96/192/336/720. Lower MSE and MAE are better. **Bold** and underlined values indicate the best and second-best distinct results at three-decimal precision, respectively. The bottom row counts top-two entries across both metrics.

Model		Ours		MixLinear		RLinear		OLinear		TimeBase		iTransformer		PatchTST	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.146**	0.198	0.173	0.226	0.171	0.223	0.147	**0.188**	0.151	0.204	0.168	0.218	0.150	0.205
	192	**0.185**	**0.233**	0.216	0.262	0.216	0.260	0.193	0.244	0.195	0.248	0.210	0.255	0.194	0.242
	336	**0.238**	**0.275**	0.262	0.297	0.261	0.294	0.247	0.276	0.244	0.282	0.260	0.291	0.242	0.283
	720	**0.312**	**0.329**	0.327	0.343	0.323	0.339	0.320	0.333	0.317	0.336	0.331	0.341	0.314	0.332
Traffic	96	0.355	0.252	0.444	0.303	0.395	0.272	**0.341**	**0.221**	0.392	0.259	0.367	0.272	0.360	0.253
	192	0.378	0.265	0.483	0.345	0.407	0.276	**0.368**	**0.233**	0.413	0.274	0.382	0.269	0.379	0.256
	336	**0.393**	0.275	0.524	0.362	0.416	0.282	0.396	**0.242**	0.427	0.287	0.395	0.279	0.397	0.264
	720	**0.429**	0.290	0.572	0.398	0.454	0.302	0.491	**0.274**	0.466	0.301	0.431	0.290	0.432	0.292
Electricity	96	0.131	0.227	0.156	0.252	0.139	0.244	0.133	**0.220**	0.136	0.229	0.132	0.228	**0.129**	0.222
	192	0.148	0.242	0.179	0.275	0.150	0.243	0.154	0.244	0.159	0.255	0.153	0.248	**0.147**	**0.240**
	336	**0.162**	**0.258**	0.202	0.295	0.166	0.260	0.168	0.260	0.172	0.288	0.168	0.265	0.163	0.259
	720	0.204	0.291	0.238	0.324	0.212	0.300	**0.194**	**0.283**	0.219	0.301	0.195	0.286	0.197	0.290
ETTh1	96	**0.359**	**0.389**	0.398	0.415	0.367	0.394	0.366	0.400	0.384	0.392	0.399	0.425	0.370	0.400
	192	**0.395**	**0.412**	0.454	0.454	0.402	0.422	0.407	0.427	0.432	0.462	0.426	0.442	0.412	0.428
	336	0.422	**0.431**	0.466	0.466	**0.419**	0.617	0.432	0.444	0.445	0.473	0.457	0.464	0.421	0.439
	720	**0.442**	**0.459**	0.443	0.467	0.451	0.463	0.452	0.470	0.449	0.482	0.630	0.574	0.447	0.467
ETTh2	96	**0.272**	0.339	0.291	0.346	0.275	0.341	0.277	**0.337**	0.401	0.439	0.297	0.356	0.279	0.341
	192	**0.333**	0.384	0.340	0.380	0.336	**0.371**	0.351	0.383	0.451	0.467	0.377	0.405	0.341	0.383
	336	0.363	0.408	0.367	0.402	**0.324**	0.385	0.370	0.402	0.458	0.482	0.424	0.440	0.328	**0.383**
	720	0.398	0.435	0.389	0.428	0.415	0.445	0.408	0.437	0.502	0.510	0.438	0.461	**0.379**	**0.422**
ETTm1	96	0.289	**0.336**	0.325	0.365	0.310	0.351	**0.279**	0.338	0.314	0.356	0.311	0.365	0.294	0.346
	192	**0.331**	**0.359**	0.351	0.381	0.338	0.367	0.334	0.362	0.339	0.370	0.348	0.385	0.333	0.370
	336	**0.364**	**0.379**	0.384	0.401	0.369	0.385	0.367	0.387	0.374	0.392	0.379	0.405	0.370	0.392
	720	0.423	**0.411**	0.435	0.428	0.429	0.415	**0.407**	0.417	0.424	0.425	0.443	0.444	0.425	0.419
ETTm2	96	**0.160**	**0.249**	0.169	0.257	0.163	0.251	0.162	0.250	0.167	0.257	0.178	0.272	0.166	0.256
	192	**0.216**	**0.287**	0.223	0.293	0.217	0.292	0.219	**0.287**	0.221	0.293	0.241	0.315	0.223	0.296
	336	**0.268**	**0.323**	0.276	0.328	0.271	0.326	0.272	0.328	0.273	0.327	0.290	0.344	0.273	0.329
	720	0.363	0.385	0.363	0.383	0.360	0.387	**0.345**	**0.372**	0.368	0.389	0.376	0.397	0.361	0.385
Count		43		5		16		25		1		5		19	

Table 2: Forecasting performance of ALTTS with alternating optimization (w/ AO) and joint training (w/o AO). Each cell reports MSE/MAE; bold denotes the lower value per metric, including ties.

(a) ETTh1, ETTh2, ETTm1, ETTm2

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	**0.359/0.389**	0.437/0.453	**0.272/0.339**	0.321/0.381	**0.289/0.336**	0.316/0.364	**0.160/0.249**	0.175/0.263
192	**0.395/0.412**	0.471/0.473	**0.333/0.384**	0.371/0.414	**0.331/0.359**	0.359/0.387	**0.216/0.287**	0.243/0.313
336	**0.422/0.431**	0.472/0.479	**0.363/0.408**	0.374/0.426	**0.364/0.379**	0.392/0.406	**0.268/0.323**	0.289/0.341
720	**0.442/0.459**	0.510/0.506	**0.398/0.435**	0.440/0.463	**0.423/0.411**	0.449/0.433	**0.363/0.385**	0.381/0.401

(b) Electricity, Traffic, Weather

Horizon (steps)	Electricity		Traffic		Weather	
	w/ AO	w/o AO	w/ AO	w/o AO	w/ AO	w/o AO
96	**0.131/0.227**	0.137/0.233	**0.355/0.252**	0.364/0.258	**0.146/0.198**	0.157/0.211
192	**0.148/0.242**	0.161/0.256	**0.378/0.265**	0.386/0.269	**0.185/0.233**	0.203/0.253
336	**0.162/0.258**	0.179/0.274	**0.393**/0.275	0.396/**0.274**	**0.238/0.275**	0.249/0.288
720	0.204/**0.291**	**0.198/0.291**	**0.429/0.290**	0.481/0.301	**0.312/0.329**	0.321/0.339

4 EXPERIMENTS

4.1 EXPERIMENTAL OVERVIEW

In this section, we benchmark ALTTS against state-of-the-art baselines to assess its forecasting accuracy across datasets and horizons. We then examine the contributions of its dual-path architecture and optimization strategy through ablation studies, and discuss its computational requirements.

Table 3: **Inference-time branch removal.** Each cell reports MSE/MAE averaged equally over horizons $\{96, 192, 336, 720\}$. Bold denotes the lowest value per metric, including ties.

Dataset	Ours	w/o CR	w/o AR
Weather	**0.2203/0.2588**	0.2238/0.2615	0.3023/0.3398
Traffic	**0.3888/0.2705**	0.4640/0.3248	1.2190/0.7505
Electricity	**0.1613/0.2545**	0.1675/0.2623	0.8463/0.7598
ETTh1	**0.4045/0.4228**	0.4060/0.4243	0.7010/0.5815
ETTh2	0.3415/0.3915	**0.3390/0.3893**	0.3903/0.4288
ETTm1	**0.3518/0.3713**	**0.3518**/0.3715	0.6930/0.5525
ETTm2	**0.2518/0.3110**	0.2533/0.3123	0.3438/0.3828

Table 4: **Sensitivity to the alternating schedule.** Results on ETTh1 with input length 512 and prediction horizon 96 using ordinary alternating updates. Rows and columns specify the AR and CR block lengths. Each cell reports MSE/MAE; bold denotes the lowest value per metric.

$N_A \backslash N_C$	1	2	5	10
1	0.3707/0.4010	0.3710/0.4017	0.3743/0.4059	0.3815/0.4120
2	0.3644/0.3952	0.3658/0.3966	0.3695/0.4006	0.3750/0.4069
5	0.3611/0.3920	0.3625/0.3934	0.3656/0.3962	0.3694/0.3998
10	0.3608/0.3924	**0.3606/0.3916**	0.3632/0.3947	0.3655/0.3963

4.2 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. Detailed dataset descriptions and statistics are provided in Appendix A.1.

Baselines. We benchmark against representative methods spanning different modeling paradigms: *RLinear* (Li et al., 2023) and *MixLinear* (Ma et al., 2026) (linear forecasters), *PatchTST* (Nie et al., 2023) (channel-independent patch-based Transformer), *iTransformer* (Liu et al., 2023) (variable-token attention for cross-variable modeling), *TimeBase* (Huang et al., 2025) (autoregression-centric backbone), and *OLinear* (Yue et al., 2025), a linear forecaster operating in an orthogonally transformed domain. These baselines cover both autoregressive and cross-variable approaches, providing a representative set of alternatives to evaluate our framework.

Implementation Details. We illustrate our *model-agnostic* optimization strategy with a *simple but effective* pairing of RLinear (Li et al., 2023) and a masked iTransformer-style encoder (Liu et al., 2023). Both branches share RevIN-normalized inputs, with their outputs summed before a single denormalization.

Training uses staged residual initialization followed by block-relaxed AO with separate Adam optimizers and AMSGrad disabled. Alternating configurations use 10 AR passes followed by 2 CR passes per cycle. We commit full AR proposals ($\alpha_A = 1$) and control CR adaptation through its learning rate and relaxation coefficient.

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.

4.3 Main Results

Table 1 summarizes results across all datasets and horizons. **Overall, AltTS attains best or second-best performance in the majority of evaluated entries.** Quantitatively, across all datasets, horizons, and both metrics (56 evaluation slots), AltTS achieves the most top-two finishes among the compared

methods, with 43 entries (32 best + 11 second-best). The closest competitor, OLinear, attains 25 top-two results, followed by PatchTST with 19. These counts use the displayed three-decimal values, with shared ties counted for each method.

On the *ETT family*, ALTTS is consistently strong on ETTh1, achieving the best MAE at all horizons and the best MSE at 96, 192, and 720. On ETTm2, it achieves the best results on both metrics at 96, 192, and 336, including a shared best MAE at 192. On ETTh2, it achieves the best MSE at 96 and 192, while other methods lead at longer horizons. On ETTm1, it achieves the best MAE at all horizons, the best MSE at 192 and 336, and the second-best MSE at 96 and 720.

On *Weather*, ALTTS achieves the best MSE at all horizons and the best MAE at 192, 336, and 720, with second-best MAE at 96. On *Electricity*, it achieves the best results on both metrics at 336, second-best MSE at 96, and second-best results on both metrics at 192; OLinear leads on both metrics at 720. On *Traffic*, ALTTS achieves the best MSE at 336 and 720 and the second-best MSE at 96 and 192. Its MAE is second-best at 96 and jointly second-best at 720, while OLinear leads on MAE at all horizons. Taken together, these results demonstrate the competitive forecasting accuracy of ALTTS.

4.4 ABLATION STUDIES

We examine branch contributions, compare forecasting performance under alternating and joint training, and assess sensitivity to the alternating schedule.

Comparison with Joint Training. Table 2 compares ALTTS with a joint-training baseline. ALTTS achieves lower MSE in 27 of 28 settings and lower MAE in 26 settings, with one MAE tie. Improvements hold on both metrics across all ETT and Weather horizons. For example, on ETTh1 at horizon 720, MSE/MAE decreases from 0.510/0.506 to 0.442/0.459. Traffic improves on MSE at every horizon, with a small MAE increase at horizon 336. On Electricity, both metrics improve at horizons 96, 192, and 336; at horizon 720, MSE increases and MAE ties. Overall, ALTTS achieves better forecasting accuracy across most evaluated settings.

Contribution of each prediction branch. We remove one prediction branch at inference time without retraining the remaining parameters. Table 3 shows that removing AR substantially increases both errors on every dataset. Removing CR has a smaller, dataset-dependent effect: the full model achieves the lowest average errors on six datasets, including an MSE tie on ETTm1, while ETTh2 improves without CR. These results support a strong temporal predictor supplemented by a cross-variable correction whose utility depends on the forecasting setting.

Sensitivity to the alternating schedule. Table 4 examines $N_A, N_C \in \{1, 2, 5, 10\}$ on ETTh1 with input length 512 and prediction horizon 96, using ordinary alternating updates. Increasing the AR block length consistently improves MSE and generally improves MAE, whereas longer CR blocks generally degrade forecasting accuracy. The 10:2 schedule achieves the lowest MSE with competitive MAE. These results support prioritizing AR updates in this setting and motivate the default alternating schedule.

Computational Considerations. The dual-path model has an additive parameter count, and the attention mask introduces no trainable parameters. AO and block relaxation operate only during training, leaving the inference graph unchanged. During an alternating block, the inactive independent branch requires only a forward evaluation. Total training cost comprises initialization and alternating updates, while inference requires one evaluation of each branch.

5 CONCLUSIONS

We introduced ALTTS, a dual-path framework that coordinates temporal and cross-variable prediction through staged alternating optimization. Our analysis characterizes gradient coupling through the shared forecasting residual, and local diagnostics illustrate its effect on temporal learning. The proposed training procedure combines compatible initialization with block relaxation to control branch adaptation. Across seven forecasting benchmarks, ALTTS achieves competitive accuracy using standard linear and attention-based modules. These findings establish branch coordination as a practical design dimension for multivariate time series forecasting.

AI USE STATEMENT

Generative AI tools assisted with literature exploration, implementation and debugging, and manuscript editing. Text polished by AI are reviewed by the authors.

ETHICS STATEMENT

This study evaluates forecasting methods on established benchmark datasets and does not involve collecting new human-subject data. Deployment in consequential applications would require application-specific validation beyond the benchmark evaluation presented here.

REPRODUCIBILITY STATEMENT

Please refer to our official implementation on anonymous github for reproducibility details.

REFERENCES

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.

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.

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.

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.

Aitian Ma, Dongsheng Luo, and Mo Sha. Mixlinear: Extreme low resource multivariate time series forecasting with 0.1k parameters. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 77764–77789, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7db2348b5bfeca620aa7327df815adcc-Paper-Conference.pdf.

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.

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.

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.

Wenzhen Yue, Yong Liu, Haoxuan Li, Hao Wang, Xianghua Ying, Ruohao Guo, Bowei Xing, and Ji Shi. Olinear: A linear model for time series forecasting in orthogonally transformed domain. *arXiv preprint arXiv:2505.08550*, 2025.

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.

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 Dataset Details

Table 5: **Dataset statistics.**

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

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 5.

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.