TS-CAN: TIME SERIES FORECASTING VIA THE CLIFFORD GEOMETRIC PRODUCT

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep time series forecasters have advanced rapidly by redesigning where representations interact, whether across patches, variables, or scales. The interaction itself, however, remains largely limited to dot-product similarities or learned weighted sums. These classical primitives solely measure how strongly two representations agree. Classical time series analysis instead suggests that relating two signals requires both an in-phase component (co-movement) and a quadrature component (oriented, out-of-phase structure). In this work, we show that the geometric product of Clifford algebra, $\mathbf{uv} = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \wedge \mathbf{v}$, yields exactly this dual decomposition within a single operation. Its generalized inner product captures scalar agreement, while its antisymmetric wedge term serves as a discrete, learnable counterpart of the Lévy area from rough path theory, capturing higher-order directional structure. Building on this principled mathematical foundation, we propose **TS-CAN**, a forecaster whose sole token interaction is the geometric product itself. Applied causally along time and sparsely across the feature dimension, this geometric interaction is representationally dense enough to internalize nonlinear channel mixing, entirely eliminating the need for self-attention and heavy feed-forward networks (FFNs). Operating at a strict linear complexity of $\mathcal{O}(N)$, TS-CAN achieves the lowest MSE in 28 of 32 long-term forecasting settings across eight benchmarks, while performing strongly on short-term tasks. These findings establish a new accuracy-efficiency Pareto frontier and suggest that how representations interact is a fundamental, unexplored design dimension for time series modeling.

1 INTRODUCTION

Time series forecasting supports decisions in energy, transportation, meteorology, and finance (Lim & Zohren, 2021; Wang et al., 2026). Deep forecasters have advanced rapidly in recent years, drawing on Transformers (Zhou et al., 2021; Wu et al., 2021; Zhou et al., 2022; Nie et al., 2023; Liu et al., 2024), MLP-based mixers (Zeng et al., 2023; Chen et al., 2023; Wang et al., 2024; 2025a), modern convolutional networks (Wu et al., 2023; Luo & Wang, 2024), and state-space models (Gu & Dao, 2023; Wang et al., 2025b). Much of this progress comes from deciding *where* representations should interact: across temporal patches (Nie et al., 2023), variables (Zhang & Yan, 2023; Liu et al., 2024), scales (Wang et al., 2024; 2025a), periods (Wu et al., 2023), and frequencies (Zhou et al., 2022). The question of *how* two representations are combined has received less attention. Most models share a common interaction primitive: a dot-product similarity, as in attention (Vaswani et al., 2017), or a learned weighted sum, as in MLPs and convolutions, usually followed by a feed-forward network (FFN). These primitives measure how strongly two representations agree, and this agreement-based view has been remarkably effective. It also raises a natural question: is agreement the only kind of relation that a forecaster needs to model between two segments of a series?

Classical time series analysis suggests that it is not. When a series is compared with a lagged or filtered reference, the relation contains two complementary components (Priestley, 1981). The in-phase component measures co-movement, while the quadrature component measures signed departure relative to the reference. The first tells how strongly the two views agree. The second distinguishes opposite orientations of change: an upward turn relative to recent context differs from an equally large downward turn, and a periodic component before its reference differs from the same component after it. The same distinction appears in nonlinear temporal analysis, where antisymmet-

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.

ric statistics characterize time irreversibility and higher-order temporal structure (Lawrance, 1991). Current deep forecasters model agreement explicitly, while such oriented structure is mostly left implicit. It is recovered indirectly through positional encodings, network depth, or FFN capacity. This suggests a complementary design principle: a forecaster could model agreement and orientation *jointly and explicitly*, within a single interaction.

Geometric (Clifford) algebra (Hestenes & Sobczyk, 1984; Dorst et al., 2007) provides exactly such an operation. The geometric product $\mathbf{uv} = \mathbf{u}\cdot\mathbf{v} + \mathbf{u}\wedge\mathbf{v}$ splits a pairwise relation into two parts. The symmetric part, the inner product, measures alignment. The antisymmetric part, the wedge product, is a bivector that encodes an oriented area and satisfies $\mathbf{u}\wedge\mathbf{v} = -\mathbf{v}\wedge\mathbf{u}$. In the plane spanned by the two vectors, these parts reduce to $\cos\varphi$ and $\sin\varphi$ of their relative angle. They therefore mirror in-phase and quadrature components: together they describe both agreement and the signed orientation of the departure from context. The wedge product also has a direct counterpart in the theory of sequential data. In rough path theory, the *Lévy area* $\frac{1}{2}\int(x\,\mathrm{d}y - y\,\mathrm{d}x)$ is the antisymmetric second-order term of the path signature (Lyons, 1998). It encodes signed area and event order in sequential paths (Flint et al., 2016; Chevyrev & Kormilitzin, 2016), and signature features have proven useful in deep sequence models (Bonnier et al., 2019). A wedge product between a token state and its shifted context is a discrete, learnable analogue of this quantity. The geometric product thus offers a single, principled primitive for combining co-movement and oriented local variation.

Building on this view, we introduce **TS-CAN**, a forecaster whose token interaction is the geometric product itself. Making this operation effective for time series requires meeting three requirements. *(i) Causality.* A directional relation is meaningful only if it respects the arrow of time. A cyclic shift along time would pair the latest patches with the earliest ones and reverse the orientation of the wrapped pairs. TS-CAN therefore pairs each state with causally shifted versions of its context, so the temporal wedge compares the present only with its own history. *(ii) Heterogeneous axes.* Direction means different things along different axes: phase and trend turning along time, and relative orientation among latent patterns within a token. TS-CAN applies separate temporal and feature-dimensional Clifford interactions, each with its own set of shifts. *(iii) Non-stationarity.* Real series drift and superimpose several periods (Kim et al., 2022; Liu et al., 2022b). TS-CAN builds a differential context that keeps only local variation, computes it with a dilated multi-scale pyramid, and complements it with a series-level mean field. For efficiency, we compute the geometric product with a sparse set of shifts, following the shifted realization of Ji (2026). The full model contains no attention and no FFN, and its cost is linear in sequence length.

We evaluate TS-CAN on long-term forecasting over eight standard benchmarks and on short-term forecasting on M4 (Makridakis et al., 2020) and PEMS. TS-CAN obtains the best or tied MSE in 28 of 32 long-term settings. On short-term forecasting, it achieves advanced or comparable performance across M4 and PEMS benchmarks. Ablations show that the two components are complementary, and a surrogate-data intervention shows that the benefit of the wedge term depends on higher-order temporal structure. Our contributions are as follows:

- We revisit the interaction primitive of deep forecasters through the lens of in-phase and oriented temporal structure. We show that the geometric product unifies agreement and signed departure, and we connect its wedge component to quadrature-like temporal asymmetry and to the Lévy area of path signatures.
- We propose TS-CAN, which realizes the geometric product along the temporal axis and the feature dimension. It uses causal shifts and a non-stationarity-aware context, contains no FFN, and runs in linear time.
- TS-CAN achieves state-of-the-art long- and short-term forecasting accuracy. A surrogate-data intervention shows that its gains from the wedge term rely on higher-order temporal structure rather than on second-order statistics or added capacity.

2 METHOD

2.1 PROBLEM FORMULATION AND OVERVIEW

Given a multivariate history $\mathbf{X} \in \mathbb{R}^{L\times M}$ with lookback length L and M variables, the goal is to predict the next F steps $\hat{\mathbf{Y}} \in \mathbb{R}^{F\times M}$. Following common practice (Kim et al., 2022; Nie et al., 2023),

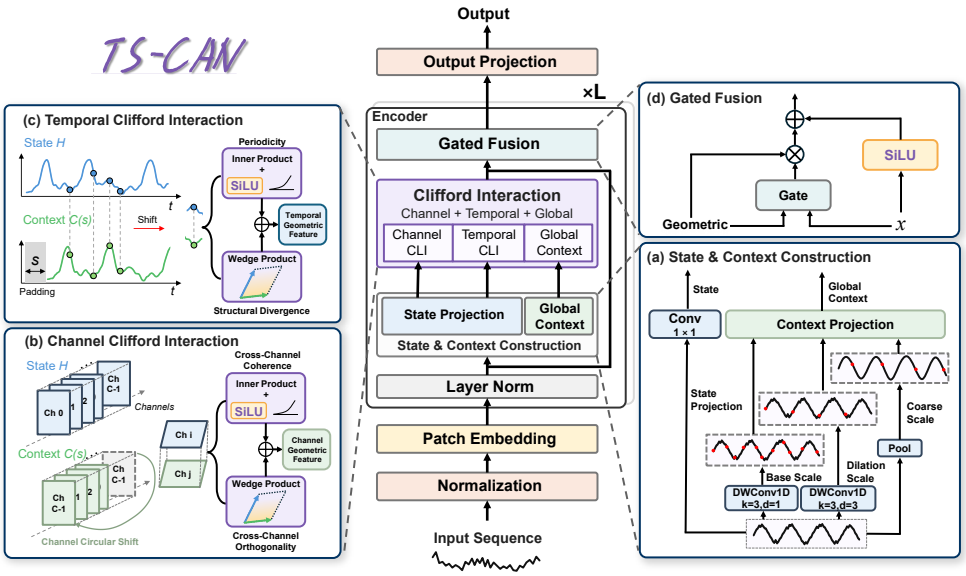


Figure 1: **Overview of TS-CAN.** After instance normalization and patch embedding, each encoder block (i) constructs a state and a multi-scale differential context (Sec. 2.2); (ii) relates them through the shifted geometric product along the channel and temporal axes, together with a global mean field (Secs. 2.3–2.4); and (iii) injects the resulting geometric features into the residual stream through gated fusion (Sec. 2.5). The block contains no attention and no feed-forward network.

we normalize each variable by its own mean and standard deviation. We then split each univariate series into N patches of length P with stride R and embed every patch into a d-dimensional token with a linear projection and a positional embedding:

$$\mathbf{Z}^0 = \mathrm{PatchEmbed}\big(\mathrm{Norm}(\mathbf{X})\big) \in \mathbb{R}^{M\times N\times d}. \tag{1}$$

Variables share parameters and are processed independently in the backbone. For brevity, we therefore write the hidden state of a single variable as $\mathbf{H} \in \mathbb{R}^{N\times d}$, whose entry $H_{n,j}$ denotes channel j of patch n. Each of the B encoder blocks updates $\mathbf{H}$ as

$$\mathbf{H}^l = \mathbf{H}^{l-1} + \boldsymbol{\gamma} \odot \mathrm{Fuse}\big(\hat{\mathbf{H}},\, \mathcal{G}(\mathbf{S}, \mathbf{C}, \bar{\mathbf{C}})\big), \qquad \hat{\mathbf{H}} = \mathrm{LN}(\mathbf{H}^{l-1}), \tag{2}$$

where $\mathbf{S}$, $\mathbf{C}$, and $\bar{\mathbf{C}}$ are the state, local context, and global context (Sec. 2.2), $\mathcal{G}$ is the Clifford interaction (Secs. 2.3–2.4), Fuse is the gated fusion (Sec. 2.5), and $\boldsymbol{\gamma} \in \mathbb{R}^d$ is a learnable LayerScale (Touvron et al., 2021). Figure 1 illustrates the full pipeline.

2.2 State and Context Construction

Every interaction in TS-CAN compares a *state*, which describes the current representation, with a *context*, which describes its surroundings. The state is a pointwise projection, $\mathbf{S} = \hat{\mathbf{H}}\mathbf{W}_s$.

Multi-scale local context. Real series often superimpose periodic patterns at several time scales. We therefore build the local context from three depthwise temporal branches: a *base* branch that stacks a standard and a dilation-2 depthwise convolution; a *dilated* branch with dilation 3; and a *coarse* branch that average-pools patches by a factor of 2, applies a pointwise projection, and linearly interpolates back to length N. Each convolutional branch is followed by BatchNorm and SiLU. The branches are combined with input-dependent weights:

$$\mathbf{C} = \sum_{i\in\{\mathrm{b,d,c}\}} \alpha_i\, \mathbf{C}_i, \qquad \boldsymbol{\alpha} = \mathrm{softmax}\Big(\mathrm{MLP}\big(\tfrac{1}{N}\textstyle\sum_n \hat{\mathbf{H}}_n\big)\Big) \in \mathbb{R}^3. \tag{3}$$

The mixture thus adapts its dominant temporal scale to each input series.

Differential context. Before interacting, we subtract the state from the context:

$$\tilde{\mathbf{C}} = \mathbf{C} - \mathbf{S}. \tag{4}$$

This subtraction removes the component that the context shares with the state and keeps only local variation. It acts as a discrete Laplacian-like high-pass filter, which makes the interaction less sensitive to level shifts in non-stationary series. Section 2.3 shows that this operation changes only the in-phase part of the interaction and leaves the directional part untouched.

Global context. To relate each patch to the series as a whole, we also compute a mean field $\bar{\mathbf{C}} = \mathbf{1}_N\big(\frac{1}{N}\sum_n \hat{\mathbf{H}}_n\big)$, which broadcasts the temporal average back to every position.

2.3 The Shifted Geometric Product

We now define the primitive shared by all interaction branches. In geometric algebra, two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ can be multiplied directly. Their *geometric product* is

$$\mathbf{uv} = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \wedge \mathbf{v}, \tag{5}$$

the sum of a scalar and a bivector. These two terms are exactly the symmetric and antisymmetric parts of the product:

$$\mathbf{u} \cdot \mathbf{v} = \tfrac{1}{2}(\mathbf{uv} + \mathbf{vu}) = \sum_i u_i v_i, \qquad \mathbf{u} \wedge \mathbf{v} = \tfrac{1}{2}(\mathbf{uv} - \mathbf{vu}) = \sum_{i<j} \big(u_i v_j - u_j v_i\big)\, \mathbf{e}_i \wedge \mathbf{e}_j. \tag{6}$$

The inner product is the familiar scalar and measures alignment. The wedge product lives in the space of bivectors. Each basis element $\mathbf{e}_i \wedge \mathbf{e}_j$ represents a unit oriented plane, and the coefficients $u_i v_j - u_j v_i$ describe the parallelogram spanned by $\mathbf{u}$ and $\mathbf{v}$, projected onto that plane. It is antisymmetric, $\mathbf{u} \wedge \mathbf{v} = -\mathbf{v} \wedge \mathbf{u}$, and satisfies $\mathbf{u} \wedge \mathbf{u} = 0$. It therefore vanishes for parallel vectors and changes sign when their order is reversed. Within the plane spanned by the two vectors, the two parts take a particularly simple form:

$$\mathbf{u} \cdot \mathbf{v} = |\mathbf{u}||\mathbf{v}| \cos\theta, \qquad \mathbf{u} \wedge \mathbf{v} = |\mathbf{u}||\mathbf{v}| \sin\theta\, \mathbf{e}_1 \wedge \mathbf{e}_2, \tag{7}$$

where θ is the signed angle from $\mathbf{u}$ to $\mathbf{v}$ and $\{\mathbf{e}_1, \mathbf{e}_2\}$ is an oriented orthonormal basis of that plane. The inner product depends on $\cos\theta$, which is even in θ, so it cannot tell whether $\mathbf{v}$ lies ahead of or behind $\mathbf{u}$. The wedge product depends on $\sin\theta$, which is odd, and resolves exactly this ambiguity. Together, the two parts determine the full relative angle. This is the geometric counterpart of the in-phase and quadrature components discussed in Section 1.

Sparse realization. Computing all $\binom{d}{2}$ bivector coefficients at every position would cost $\mathcal{O}(d^2)$. We instead sample them sparsely with a shift operator $\mathcal{T}_k$ that pairs each entry with the entry k positions away along a chosen axis, either channels or time. For a set of shifts $\mathcal{K}$, we define

$$\mathcal{I}_k(\mathbf{S}, \mathbf{C}) = \mathrm{SiLU}\big(\mathbf{S} \odot \mathcal{T}_k\mathbf{C}\big), \qquad \mathcal{W}_k(\mathbf{S}, \mathbf{C}) = \mathbf{S} \odot \mathcal{T}_k\mathbf{C} - \mathbf{C} \odot \mathcal{T}_k\mathbf{S}, \tag{8}$$

At entry j, $\mathcal{W}_k$ equals $S_j C_{j-k} - C_j S_{j-k}$. This is precisely the coefficient of the plane spanned by $\mathbf{e}_{j-k}$ and $\mathbf{e}_j$ in Eq. equation 6, up to orientation. Each shift therefore extracts one diagonal of the bivector coefficient matrix, and a set of shifts samples $|\mathcal{K}|$ such diagonals in $\mathcal{O}(|\mathcal{K}|d)$ time. Similarly, $\mathcal{I}_k$ keeps the corresponding diagonal of the outer product $\mathbf{S}\mathbf{C}^\top$, whose main diagonal sums to the standard inner product. We pass it through SiLU, which turns it into a coherence gate. The gated term is no longer strictly symmetric, so we refer to it as a gated in-phase term. All shifts are aggregated by a learnable pointwise projection $\mathcal{P} : \mathbb{R}^{2|\mathcal{K}|d} \to \mathbb{R}^d$:

$$\mathcal{G}_{\mathcal{T}}(\mathbf{S}, \mathbf{C}) = \mathcal{P}\Big(\big\|_{k\in\mathcal{K}} \mathcal{W}_k(\mathbf{S}, \mathbf{C}) \,\big\|\, \big\|_{k\in\mathcal{K}} \mathcal{I}_k(\mathbf{S}, \mathbf{C})\Big), \tag{9}$$

where $\|$ denotes concatenation along channels. This shifted realization follows Ji (2026). Since $\mathcal{P}$ is learnable, any sign flips that arise when cyclically shifted index pairs wrap around are absorbed by the projection.

The antisymmetric term has a direct interpretation for time series. Let $\mathcal{T}_k$ be the causal temporal shift, $(\mathcal{T}_k\mathbf{C})_n = \mathbf{C}_{n-k}$ with zeros for $n \le k$, and consider a single channel. Write the state and context of that channel as s_n and c_n, and view the pairs (c_n, s_n) as a path in the plane.

Proposition 1 (Discrete Lévy area). *For $k = 1$, $\mathcal{W}_1(\mathbf{s}, \mathbf{c})_n = c_{n-1}s_n - s_{n-1}c_n$ is twice the signed area swept by the path segment from (c_{n-1}, s_{n-1}) to (c_n, s_n) as seen from the origin. Consequently, $\frac{1}{2}\sum_{n=1}^{N} \mathcal{W}_1(\mathbf{s}, \mathbf{c})_n$ equals the Lévy area of the piecewise-linear path $0 \to (c_1, s_1) \to \cdots \to (c_N, s_N)$.*

The causal zero padding plays a natural role here: it anchors the path at the origin. For $k > 1$, $\mathcal{W}_k$ measures the same oriented area between a state and its k-step history, so a set of shifts yields a multi-lag profile of directional structure.

Proposition 2 (Quadrature spectrum). *If (s_n, c_n) is jointly weakly stationary with cross-covariance $\gamma_{sc}(k) = \mathrm{Cov}(s_n, c_{n-k})$, then for $n > k$*

$$\mathbb{E}\big[\mathcal{W}_k(\mathbf{s}, \mathbf{c})_n\big] = \gamma_{sc}(k) - \gamma_{sc}(-k),$$

which is twice the odd part of the lagged cross-covariance. The quadrature spectrum is, up to sign convention, $q_{sc}(\omega) = \frac{1}{2\pi}\sum_{k\geq 1}\big(\gamma_{sc}(k) - \gamma_{sc}(-k)\big)\sin(\omega k)$, so $\{\mathbb{E}[\mathcal{W}_k]\}_k$ are exactly its sine-series coefficients.

Proposition 2 makes precise the link to classical cross-spectral analysis discussed in Section 1. The wedge term is a local, learnable estimator of the out-of-phase, lead–lag component. Its expectation vanishes when the two sequences are time-reversible with respect to each other, and it changes sign when their roles are swapped. Proofs are given in Appendix C. Finally, the differential context in Eq. equation 4 interacts with the wedge term in a simple way:

Remark 1 (Invariance of the wedge term). *Since $\mathcal{W}_k(\mathbf{S}, \mathbf{S}) = \mathbf{0}$ and $\mathcal{W}_k$ is bilinear, $\mathcal{W}_k(\mathbf{S}, \mathbf{C} - \mathbf{S}) = \mathcal{W}_k(\mathbf{S}, \mathbf{C})$. The differential context therefore modifies only the in-phase term. It removes the self-energy $\mathbf{S} \odot \mathcal{T}_k\mathbf{S}$ from the in-phase term and leaves the directional structure intact.*

2.4 Clifford Interaction along Channel and Temporal Axes

The operator $\mathcal{G}_{\mathcal{T}}$ is instantiated three times. The branches differ only in the shift operator and the context they use:

$$\mathcal{G} = \mathcal{G}_{\mathcal{T}^{\mathrm{ch}}}(\mathbf{S}, \tilde{\mathbf{C}}) + \beta_t\, \mathcal{G}_{\mathcal{T}^{\mathrm{tem}}}(\mathbf{S}, \tilde{\mathbf{C}}) + \beta_g\, \mathcal{G}_{\mathcal{T}^{\mathrm{ch}}}(\mathbf{S}, \bar{\mathbf{C}}), \tag{10}$$

where β_t and β_g are learnable scalars and each branch has its own projection $\mathcal{P}$.

Channel interaction. The channel branch shifts along the feature axis and relates channel $a = j - k$ to channel $b = j$. For a patch n, consider the state and the context as two vectors in the plane spanned by $\mathbf{e}_a$ and $\mathbf{e}_b$. The wedge coefficient $S_{n,b}C_{n,a} - S_{n,a}C_{n,b}$ is the oriented area between them. It vanishes whenever the state is proportional to its context, that is, whenever the *relative composition* of the two features is unchanged, regardless of their overall magnitude. Its sign indicates which feature is gaining weight relative to the other. The channel wedge thus detects shifts in the balance between learned patterns, such as a trend-like feature giving way to an oscillatory one. This reading has a precise form when the context is a temporal filter of the state (Proposition 3): the channel wedge becomes a weighted contrast of the signed areas swept by the two-dimensional path $(S_{n,a}, S_{n,b})$ around the context window. The temporal and channel branches therefore sample complementary slices of the same object, the second-level signature of the latent path (Chevyrev & Kormilitzin, 2016). The temporal branch computes lagged areas within each feature, and the channel branch computes areas across feature pairs, which is the classical setting of the Lévy area. Because the projection $\mathcal{P}$ already mixes all channels, the reach of individual shifts is not critical. In practice, accuracy depends mainly on the number of shifts rather than on their spacing (Appendix D).

Temporal interaction. The temporal branch shifts along the patch axis and pairs each state with its own history. Here causality is essential. A cyclic shift would pair the most recent patches with the earliest ones and thus reverse the orientation of the wrapped pairs. We therefore use a right shift with zero padding. As Propositions 1 and 2 show, this branch estimates multi-lag oriented areas, which carry phase shift and turning direction. Its in-phase term captures lagged co-movement and thus reflects periodicity.

Global interaction. The global branch uses the mean field $\bar{\mathbf{C}}$ as context, so it measures how each patch deviates from the series as a whole. The mean field is constant over time, so a temporal shift carries no information here, and we apply the channel shift instead. We use only the in-phase term in this branch by default.

Table 1: Average results for the long-term forecasting task. Each value is averaged over prediction lengths $\{96, 192, 336, 720\}$. Full horizon-wise results are given in Table 15.

Dataset	**TS-CAN**		TimeMixer++		TimeMixer		iTransformer		PatchTST		Crossformer		TiDE		TimesNet		DLinear		SCINet		FEDformer		Stationary		Autoformer	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	**0.224**	**0.261**	0.226	0.262	0.240	0.271	0.258	0.278	0.265	0.285	0.264	0.320	0.271	0.320	0.259	0.287	0.265	0.315	0.292	0.363	0.309	0.360	0.288	0.314	0.338	0.382
Electricity	**0.164**	0.259	0.165	**0.253**	0.182	0.272	0.178	0.270	0.216	0.318	0.244	0.334	0.251	0.344	0.192	0.304	0.225	0.319	0.268	0.365	0.214	0.327	0.193	0.296	0.227	0.338
Traffic	**0.387**	0.276	0.416	**0.264**	0.484	0.297	0.428	0.282	0.529	0.341	0.667	0.426	0.760	0.473	0.620	0.336	0.625	0.383	0.804	0.509	0.610	0.376	0.624	0.340	0.628	0.379
Exchange	**0.338**	0.392	0.357	**0.391**	0.391	0.453	0.360	0.403	0.367	0.404	0.940	0.707	0.370	0.413	0.416	0.443	0.354	0.414	0.750	0.626	0.519	0.429	0.461	0.454	0.613	0.539
ETTh1	**0.412**	**0.430**	0.419	0.432	0.447	0.440	0.454	0.447	0.516	0.484	0.529	0.522	0.541	0.507	0.458	0.450	0.461	0.457	0.747	0.647	0.498	0.484	0.570	0.537	0.496	0.487
ETTh2	0.341	0.384	**0.339**	**0.380**	0.364	0.395	0.383	0.407	0.391	0.411	0.942	0.684	0.611	0.550	0.414	0.427	0.563	0.519	0.954	0.723	0.437	0.449	0.526	0.516	0.450	0.459
ETTm1	**0.352**	0.384	0.369	**0.378**	0.381	0.395	0.407	0.410	0.406	0.407	0.513	0.495	0.419	0.419	0.400	0.406	0.404	0.408	0.485	0.481	0.448	0.452	0.481	0.456	0.588	0.517
ETTm2	**0.258**	**0.319**	0.269	0.320	0.275	0.323	0.288	0.332	0.290	0.334	0.757	0.610	0.358	0.404	0.291	0.333	0.354	0.402	0.954	0.723	0.305	0.349	0.306	0.347	0.327	0.371

2.5 Gated Fusion and Forecasting Head

The geometric features are injected into the residual stream through a gate conditioned on both the normalized input and the geometric features:

$$\mathrm{Fuse}(\hat{\mathbf{H}}, \mathcal{G}) = \mathrm{SiLU}(\hat{\mathbf{H}}) + \mathbf{g} \odot \mathcal{G}, \qquad \mathbf{g} = \sigma\big([\hat{\mathbf{H}} \,\|\, \mathcal{G}]\, \mathbf{W}_g\big). \tag{11}$$

The SiLU path passes through a non-linearly filtered copy of the current representation, while the gate decides, per position and channel, how much geometric information to add. In the view of Eq. equation 2 as an Euler step of a feature flow, the gate acts as an adaptive step size for the geometric force. We apply stochastic depth (Huang et al., 2016) to the residual branch and initialize γ to a small value for stable optimization. After B blocks, a final LayerNorm is applied. The N tokens of each variable are then flattened and mapped to the horizon by a linear head. Finally, the output is de-normalized with the statistics from Eq. equation 1:

$$\hat{\mathbf{Y}}_{:,m} = \mathrm{DeNorm}\Big(\mathrm{Linear}\big(\mathrm{Flatten}(\mathrm{LN}(\mathbf{H}_m^B))\big)\Big). \tag{12}$$

The model is trained end-to-end with the loss prescribed by each benchmark protocol (Wu et al., 2023).

2.6 Complexity

For each branch, the shifted products in Eq. equation 8 cost $\mathcal{O}(MNd|\mathcal{K}|)$ element-wise operations, and the projection $\mathcal{P}$ costs $\mathcal{O}(MN|\mathcal{K}|d^2)$. The context construction uses depthwise convolutions with cost $\mathcal{O}(MNd)$. The total cost of a block is therefore linear in the number of patches N, compared with $\mathcal{O}(N^2)$ for self-attention along time and $\mathcal{O}(M^2)$ for attention across variables. All operations are dense tensor shifts and pointwise products, with no irregular memory access. We report measured parameters, FLOPs, memory, and runtime in Section 3.5.

3 Experiments

3.1 Experimental Setup

We evaluate TS-CAN on long-term forecasting over eight standard benchmarks and on short-term forecasting over M4 and PEMS. We follow the Time-Series-Library protocol (Wu et al., 2023; Wang et al., 2026), compare against recent MLP-, Transformer-, and convolution-based forecasters, and report MSE/MAE for long-term forecasting, SMAPE/MASE/OWA for M4, and MAE/MAPE/RMSE for PEMS. TS-CAN is implemented in PyTorch with MSE loss, Adam, and mixed-precision training. Full dataset, baseline, protocol, and hyperparameter details are provided in Appendices C.6 and C.7.

3.2 Main Results

Long-term forecasting. Table 1 reports the average results, and the complete horizon-wise results are provided in Appendix F. TS-CAN achieves the lowest MSE in 28 of the 32 dataset–horizon settings (ties included) and the lowest average MSE on seven of the eight datasets. The largest gains appear on the benchmarks where directional structure matters most. On Traffic, whose 862 sensors exhibit strong daily and weekly cycles with sensor-specific phase offsets, TS-CAN reduces the average MSE by 7.0% relative to the strongest baseline (0.387 vs. 0.416). On the highly non-stationary

Table 2: Short-term forecasting results averaged over the four M4 subsets. Lower SMAPE, MASE, and OWA indicate better prediction. **Red**: best; Blue: second best.

Metric	**TS-CAN**	TimeMixer++	TimeMixer	iTransformer	TiDE	TimesNet	N-HiTS	N-BEATS*	PatchTST	MICN	FiLM	LightTS	DLinear	FEDformer	Stationary	Autoformer
SMAPE	**9.965**	10.016	10.093	11.011	12.112	10.262	10.368	10.290	11.766	24.791	13.224	13.876	12.333	10.934	11.223	11.189
MASE	1.998	**1.982**	2.029	2.380	2.618	2.103	2.106	2.135	2.697	18.325	3.024	4.231	2.948	2.174	2.391	2.367
OWA	**0.859**	0.871	0.863	0.995	1.095	0.897	0.908	0.903	1.075	4.663	1.233	1.571	1.152	0.952	1.022	1.006

* Following the TimeMixer++ comparison, N-BEATS is evaluated without its ensemble.

Table 3: Ablation of the geometric product and the interaction axes on ETTh1 ($F = 96$). Δ is the relative change in MSE with respect to the full model. The unstructured bilinear variant feeds the raw products $[\mathbf{S} \odot \mathcal{T}_k\mathbf{C},\ \mathbf{C} \odot \mathcal{T}_k\mathbf{S}]$ to the same projection $\mathcal{P}$, without subtraction or SiLU, in both the channel and temporal branches.

(a) Components and parameterization

Variant	Params	MSE	MAE	Δ
Full (inner + wedge)	1.26M	**0.361**	**0.393**	–
Channel branch				
inner only	1.10M	0.382	0.402	+6.1%
wedge only	1.10M	0.381	0.400	+5.6%
Temporal branch				
inner only	1.10M	0.380	0.402	+5.4%
wedge only	1.10M	0.380	0.401	+5.4%
Parameterization				
unstructured bilinear	1.26M	0.369	0.395	+2.4%

(b) Interaction axes and temporal shift

Variant	Params	MSE	MAE	Δ
Full	1.26M	**0.361**	**0.393**	–
w/o channel branch	0.93M	0.378	0.405	+4.9%
w/o temporal branch	0.93M	0.371	0.395	+2.9%
w/o global branch	1.10M	0.377	0.400	+4.4%
circular temporal shift	1.26M	0.362	0.394	+0.4%

Exchange dataset, the gain grows with the horizon and reaches 6.9% over TimeMixer++ at $F = 720$. TS-CAN also improves the average MSE on ETTm1 and ETTm2 by 4.6% and 4.1%. Its advantage is smaller in MAE, where TimeMixer++ remains the strongest competitor. Because MSE penalizes large errors more heavily than MAE, this pattern suggests that TS-CAN is particularly effective at reducing large deviations. ETTh2 is the only dataset where TS-CAN trails TimeMixer++, by 0.002 in average MSE.

Short-term forecasting. Table 2 summarizes the average result over the four M4 subsets, with complete M4 and PEMS results deferred to Appendix E. TS-CAN achieves the best average SMAPE and OWA on M4, and the second-best average MASE. Together with the detailed PEMS comparison in the appendix, these results show that the same interaction primitive transfers beyond long-horizon multivariate forecasting without task-specific modules.

3.3 Ablation Studies

We ablate TS-CAN on ETTh1 with horizon 96 and change one factor at a time. Each ablation result is averaged over three random seeds. Tables 3 and 4 report MSE, MAE, the relative change in MSE, and the number of parameters.

Both components of the geometric product are necessary. Table 3(a) keeps only one component in either the channel or the temporal branch. Every such variant increases MSE by 5.4–6.1%. The inner-only and wedge-only variants perform almost identically. The wedge term carries no self-energy, since $\mathbf{u} \wedge \mathbf{u} = 0$, yet it is as informative as the in-phase term. Neither component alone comes close to the full model. This matches the spectral view of Proposition 2: the two components carry complementary information, namely co-movement and directional structure, and neither can replace the other.

Explicit antisymmetry outperforms an unstructured bilinear interaction. Replacing $[\mathcal{W}_k, \mathcal{I}_k]$ with the raw products $[\mathbf{S} \odot \mathcal{T}_k\tilde{\mathbf{C}},\ \tilde{\mathbf{C}} \odot \mathcal{T}_k\mathbf{S}]$, without subtraction or SiLU, keeps the parameter count identical and still allows $\mathcal{P}$ to represent the wedge exactly. Yet it raises MSE by 2.4% (Table 3a). The explicit wedge cancels the self-interaction $\mathbf{S} \odot \mathcal{T}_k\mathbf{S}$ shared by both products (Remark 1), a cancellation that the unconstrained projection does not learn from finite data. Although this variant also drops the SiLU gate, the result shows that the gain is not explained by a generic multiplicative interaction of equal capacity.

Table 4: Ablation of context construction, interaction primitive, and fusion on ETTh1 ($F = 96$). "Fixed" uses uniform weights across scales instead of the adaptive weights of Eq. equation 3.

(a) Context construction

Variant	Params	MSE	MAE	Δ
Full (diff, 3 scales, adapt.)	1.26M	**0.361**	**0.393**	–
absolute context	1.26M	0.362	0.393	+0.2%
base only	1.22M	0.378	0.401	+4.9%
base + dilated, fixed	1.22M	0.383	0.403	+6.1%
base + dilated, adapt.	1.23M	0.365	0.396	+1.1%
3 scales, fixed	1.25M	0.372	0.400	+3.1%

(b) Interaction primitive and fusion

Variant	Params	MSE	MAE	Δ
TS-CAN (no FFN)	1.26M	**0.361**	**0.393**	–
self-attention + FFN	1.26M	0.388	0.409	+7.7%
MLP mixer + FFN	1.26M	0.374	0.396	+3.6%
DW-sep. conv + FFN	1.26M	0.379	0.398	+5.1%
TS-CAN + FFN	1.36M	0.371	0.396	+2.7%
direct-add fusion	1.26M	0.361	0.395	+0.0%
w/o SiLU path	1.26M	0.364	0.393	+0.9%

Table 5: Surrogate intervention. Wedge gain is the relative MSE reduction of the full model over the in-phase-only model ($F = 96$); positive means the wedge term helps. IAAFT surrogates preserve each variable's amplitude distribution and power spectrum while randomizing higher-order temporal structure. The last column is the irreversibility index (Appendix C.5) before and after the surrogate transform.

Dataset	Wedge gain (original)	Wedge gain (surrogate)	Irreversibility
ETTh1	**+5.2%**	−0.6%	$0.222 \to 0.081$
ETTm1	**+3.4%**	+0.1%	$0.262 \to 0.115$
ETTm2	+0.5%	+0.7%	$0.419 \to 0.059$
ETTh2	−2.2%	−4.6%	$0.477 \to 0.108$
Average	+1.7%	−1.1%	–

Every interaction axis contributes. As shown in Table 3(b), removing any branch degrades accuracy. The channel branch contributes most (+4.9%), followed by the global (+4.4%) and temporal (+2.9%) branches. The three axes therefore provide complementary rather than redundant information. Replacing the causal temporal shift with a cyclic one slightly increases the error. All patches lie within the lookback window, so the cyclic shift does not expose any target information. It does, however, pair the most recent patches with the earliest ones, which reverses the orientation of the wrapped pairs. The causal shift keeps a consistent past-to-present orientation and completes the Lévy-area interpretation of Proposition 1. It is also marginally more accurate.

Multi-scale, adaptive context matters; differencing affects only the in-phase term. Table 4(a) shows that restricting the context to the base scale increases MSE by 4.9%. The adaptive weights are essential. With fixed uniform weights, adding scales helps little or even hurts (+6.1% for two scales, +3.1% for three). With adaptive weights, each additional scale gives a steady improvement, which indicates that the model benefits from selecting its dominant temporal scale per series. Switching from the differential to the absolute context changes the error by only 0.2%. This is consistent with Remark 1: differencing leaves the wedge term invariant and only removes the self-energy from the in-phase term, so its effect is expected to be modest on a single benchmark. We keep the differential context for its better behavior on the non-stationary datasets.

The interaction primitive matters more than capacity. In Table 4(b), we keep the full pipeline, including the patching, context, fusion, and head, and replace only the Clifford interaction with a standard token mixer followed by an FFN, using parameter-matched configurations. Self-attention, MLP, and depthwise-separable convolution mixers increase MSE by 7.7%, 3.6%, and 5.1%. The gap under similar parameter counts indicates that the gain comes from the geometric interaction itself rather than simply from model size. Appending an FFN to TS-CAN adds 0.1M parameters and increases MSE by 2.7%, which further suggests that the geometric interaction already provides much of the transformation capacity that FFNs usually supply. Finally, the fusion design has a minor influence. Direct addition matches TS-CAN in MSE but is slightly worse in MAE. We adopt the gated form for its lower MAE and more stable training across datasets.

Table 6: Efficiency comparison under a unified setting (lookback 96, horizon 96, batch size 16, single RTX 4090D). Avg. MSE is averaged over all eight datasets and four horizons of Table 1. Full results, including FLOPs, are given in Table 9.

Model	Params (M)	Train mem. (GB)	Train (ms/iter)	Infer. (ms/batch)	Avg. MSE
TS-CAN	0.950	**0.017**	4.10	<u>0.32</u>	**0.310**
TimeMixer	0.207	0.280	13.99	2.45	0.346
iTransformer	0.290	0.024	<u>3.27</u>	0.66	<u>0.345</u>
PatchTST	0.415	0.049	3.54	0.79	0.373
TimesNet	37.514	18.726	203.80	22.29	0.381
Crossformer	2.013	0.112	23.46	3.94	0.607
FEDformer	0.865	0.136	130.56	27.12	0.418
DLinear	0.019	**0.017**	**0.69**	**0.11**	0.406

3.4 Does the Wedge Term Exploit Directional Structure?

If the wedge term captures directional structure, its benefit should disappear once that structure is removed, even when everything the in-phase term can use is kept intact. We test this prediction with IAAFT surrogates (Schreiber & Schmitz, 1996). These surrogates preserve each variable's amplitude distribution and power spectrum, and hence all second-order statistics. At the same time, they randomize higher-order temporal structure, including time irreversibility (Lawrance, 1991): the irreversibility index drops by 56–86%. We train in-phase-only and full models on the original and surrogate versions of the four ETT datasets using the same training protocol as the main experiments, and report averages over three random seeds (Table 5). On ETTh1 and ETTm1, where the wedge term helps most on the original data (+5.2% and +3.4%), the gain vanishes on the surrogates (-0.6% and $+0.1\%$). On ETTm2, the gain is negligible in both settings. On ETTh2, the wedge term does not help even on the original data. This is consistent with ETTh2 being the only dataset where TS-CAN trails the best baseline in Table 1. Averaged over the four datasets, the gain drops from +1.7% to -1.1%. The full model has more parameters than the in-phase-only model, yet this extra capacity brings no benefit on the surrogates. Where the wedge term improves forecasting, the improvement therefore depends on higher-order temporal structure that second-order statistics cannot capture, rather than on added capacity. We also note that the irreversibility index alone does not rank datasets by gain. ETTh2 and ETTm2 are the most irreversible, yet benefit least, likely because their asymmetry is dominated by sporadic spikes that are hard to predict. An operator-level verification of Corollary 3 is provided in Appendix C.5.

3.5 Efficiency

Table 6 compares training and inference costs with all models set to the same width and input size (Appendix C.7). TS-CAN attains the lowest average MSE while remaining cheap to run: it ties DLinear for the lowest training memory, has the second-fastest inference latency, and trains at a speed comparable to iTransformer and PatchTST. Compared with TimeMixer, TS-CAN trains 3.4× faster, runs inference 7.7× faster, and uses 16× less training memory. Although its pointwise projections add parameters, they map to dense matrix multiplications, while the remaining operations are shifts and element-wise products without attention or softmax. Thus the extra parameters do not translate into high latency or memory, and the shift set $\mathcal{K}$ provides a simple size–accuracy knob.

4 Conclusion

We revisited the interaction primitive of deep forecasters from the perspective of classical time series analysis. The Clifford geometric product gives a single operation for modeling both co-movement, through its in-phase term, and oriented temporal departure, through its wedge term, a discrete counterpart of the Lévy area whose expectations recover quadrature-like structure. TS-CAN applies this primitive along the temporal axis and feature dimension without attention or feed-forward networks, achieving strong long- and short-term forecasting accuracy with low memory and latency. Ablations and surrogate-data interventions further show that the two components are complementary and that the wedge gain depends on higher-order temporal structure.

AI USE STATEMENT

We used generative AI tools to assist with manuscript polishing, wording refinement, LaTeX table formatting, and consistency checks across sections. We also used AI assistance to help run and organize repeated experiments, including checking commands, logs, and result tables. The research idea, model design, experimental conclusions, and final claims were reviewed and validated by the authors. The authors take full responsibility for the final content of this submission.

ETHICS STATEMENT

This work studies general-purpose time series forecasting methods. All datasets used in the experiments are publicly available benchmark datasets, and we do not collect private data or conduct studies involving human subjects. We are not aware of direct ethical risks beyond those normally associated with forecasting models, such as potential misuse when forecasts are deployed without uncertainty assessment or domain oversight.

REPRODUCIBILITY STATEMENT

We describe the model architecture and computational complexity in Sections 2 and 2.6. Experimental protocols, datasets, baselines, and implementation details are summarized in Section 3 and expanded in Appendix C.7. The appendix also provides complete tables, ablation settings, surrogate-data procedures, and proofs for the theoretical claims.

REFERENCES

Patric Bonnier, Patrick Kidger, Imanol Perez Arribas, Cristopher Salvi, and Terry Lyons. Deep signature transforms. In *Advances in Neural Information Processing Systems*, volume 32, pp. 3105–3115, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/d2cdf047a6674cef251d56544a3cf029-Abstract.html.

Johannes Brandstetter, Rianne van den Berg, Max Welling, and Jayesh K. Gupta. Clifford neural layers for PDE modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=okwxL_c4x84.

Johann Brehmer, Pim de Haan, Sonke Behrends, and Taco Cohen. Geometric algebra transformer. In *Advances in Neural Information Processing Systems*, volume 36, pp. 35472–35496, 2023. doi: 10.52202/075280-1541. URL https://openreview.net/forum?id=M7r2CO4tJC.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler-Canseco, and Artur Dubrawski. NHITS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023. doi: 10.1609/aaai.v37i6.25854.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecasting. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=wbpxTuXgm0.

Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. *arXiv preprint arXiv:1603.03788*, 2016. URL https://arxiv.org/abs/1603.03788.

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. URL https://openreview.net/forum?id=pCbC3aQB5W.

Leo Dorst, Daniel Fontijne, and Stephen Mann. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*. Morgan Kaufmann, Amsterdam, 2007.

Guy Flint, Ben Hambly, and Terry Lyons. Discretely sampled signals and the rough hoff process. *Stochastic Processes and their Applications*, 126(9):2593–2614, 2016. doi: 10.1016/j.spa.2016.02.011.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. URL https://arxiv.org/abs/2312.00752.

Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):7129–7142, 2024. doi: 10.1109/TKDE.2024.3400008.

David Hestenes and Garret Sobczyk. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. D. Reidel Publishing Company, Dordrecht, 1984.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In *Computer Vision – ECCV 2016*, pp. 646–661. Springer, 2016. doi: 10.1007/978-3-319-46493-0_39.

Zhongping Ji. CliffordNet: All you need is geometric algebra. *arXiv preprint arXiv:2601.06793*, 2026. doi: 10.48550/arXiv.2601.06793. URL https://arxiv.org/abs/2601.06793.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6696–6707, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4a5876b450b45371f6cfe5047ac8cd45-Abstract.html.

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*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 95–104, 2018. doi: 10.1145/3209978.3210006.

A. J. Lawrance. Directionality and reversibility in time series. *International Statistical Review / Revue Internationale de Statistique*, 59(1):67–79, 1991. doi: 10.2307/1403575.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379 (2194):20200209, 2021. doi: 10.1098/rsta.2020.0209.

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 *Advances in Neural Information Processing Systems*, volume 35, pp. 5816–5828, 2022a. doi: 10.52202/068431-0421. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/266983d0949aed78a16fa4782237dea7-Abstract-Conference.html.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022b. URL https://openreview.net/forum?id=ucNDIDRNjjv.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Donghao Luo and Xue Wang. ModernTCN: A modern pure convolution structure for general time series analysis. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vpJMJerXHU.

Terry J. Lyons. Differential equations driven by rough signals. *Revista Matem 'atica Iberoamericana*, 14(2):215–310, 1998. doi: 10.4171/RMI/240.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1): 54–74, 2020. doi: 10.1016/j.ijforecast.2019.04.014.

James Morrill, Cristopher Salvi, Patrick Kidger, and James Foster. Neural rough differential equations for long time series. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7829–7838. PMLR, 2021. URL https://proceedings.mlr.press/v139/morrill21b.html.

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. URL https://openreview.net/forum?id=Jbdc0vTOcol.

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. URL https://openreview.net/forum?id=r1ecqn4YwB.

Maurice B. Priestley. *Spectral Analysis and Time Series*. Probability and Mathematical Statistics. Academic Press, London and New York, 1981.

David Ruhe, Johannes Brandstetter, and Patrick Forre. Clifford group equivariant neural networks. In *Advances in Neural Information Processing Systems*, volume 36, pp. 62922–62990, 2023. doi: 10.52202/075280-2748. URL https://openreview.net/forum?id=n84bzMrGUD.

Thomas Schreiber and Andreas Schmitz. Improved surrogate data for nonlinearity tests. *Physical Review Letters*, 77(4):635–638, 1996. doi: 10.1103/PhysRevLett.77.635.

James Theiler, Stephen Eubank, André Longtin, Bryan Galdrikian, and J. Doyne Farmer. Testing for nonlinearity in time series: The method of surrogate data. *Physica D: Nonlinear Phenomena*, 58(1–4):77–94, 1992. doi: 10.1016/0167-2789(92)90102-S.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021. doi: 10.1109/ICCV48922.2021.00010.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

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 *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=1CLzLXSFNn.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(10):11642–11662, 2026. doi: 10.1109/TPAMI.2026.3690845.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025b. doi: 10.1016/j.neucom.2024.129178.

Gideon Weiss. Time-reversibility of linear stochastic processes. *Journal of Applied Probability*, 12 (4):831–836, 1975. doi: 10.2307/3212735.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

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. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with 10k parameters. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bWcnvZ3qMb.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 76656–76679, 2023. doi: 10.52202/075280-3349.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JiTVtCUOpS.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

A Related Work

A.1 Deep Time Series Forecasting

Deep forecasters have evolved through several architectural families. Transformer-based models introduced sparse attention (Zhou et al., 2021), series decomposition with auto-correlation (Wu et al., 2021), and frequency-enhanced blocks (Zhou et al., 2022). Later work showed that patching (Nie et al., 2023) and treating variables as tokens (Liu et al., 2024) substantially improve accuracy. Simple linear and MLP-based models proved highly competitive (Zeng et al., 2023; Chen et al., 2023). Multi-scale mixing (Wang et al., 2024; 2025a), period-aware 2D convolutions (Wu et al., 2023), modern large-kernel convolutions (Luo & Wang, 2024), and state-space models (Gu & Dao, 2023; Wang et al., 2025b) further broadened the design space. Another line of work studies dependencies between variables. It covers channel-independent versus channel-mixing strategies (Nie et al., 2023; Han et al., 2024), cross-dimension attention (Zhang & Yan, 2023), and explicit modeling of leading indicators (Zhao & Shen, 2024). Phase information has mainly been handled in the frequency domain, through Fourier-based attention or complex-valued spectral layers (Zhou et al., 2022; Yi et al., 2023; Xu et al., 2024). Most of these works define *where* representations interact, across

patches, variables, scales, or frequencies, and typically rely on a similarity-based or linear primitive. TS-CAN is complementary. It keeps a standard patch-based pipeline and changes the interaction primitive itself, modeling agreement and direction jointly and locally in the representation space rather than globally in the spectral domain.

A.2 Geometric Algebra and Path Signatures in Deep Learning

Geometric (Clifford) algebra (Hestenes & Sobczyk, 1984; Dorst et al., 2007) has recently been adopted in deep learning, mainly for physical systems. Clifford neural layers represent fields as multivectors for PDE modeling (Brandstetter et al., 2023). Clifford group equivariant networks (Ruhe et al., 2023) and the Geometric Algebra Transformer (Brehmer et al., 2023) build E(n)-equivariant architectures for geometric data. These models use multivectors as the data type and Clifford-valued weights, which suits 3D and physical inputs. More recently, Ji (2026) used the geometric product as a token interaction for image recognition and proposed an efficient shifted realization, which we adopt. A related line of work comes from rough path theory. Path signatures summarize sequential data through iterated integrals (Lyons, 1998; Chevyrev & Kormilitzin, 2016). Their antisymmetric second-order term, the Lévy area, captures event order and lead–lag structure (Flint et al., 2016). Signatures have been used as features or layers in deep models (Bonnier et al., 2019), and they underpin neural controlled and rough differential equations (Kidger et al., 2020; Morrill et al., 2021). TS-CAN connects these two lines. Instead of computing truncated signatures, whose size grows rapidly with channel dimension and depth, it learns a sparse, discrete counterpart of the Lévy area directly inside the backbone through the wedge product. It also adds the causal temporal interactions, channel interactions, and non-stationarity-aware context that forecasting requires.

B Limitations and Future Work

TS-CAN is less accurate than the strongest baseline on ETTh2, and its advantage is smaller in MAE than in MSE. Its parameter count, which comes mainly from the pointwise projections, is higher than that of the most lightweight baselines. Our surrogate analysis covers the ETT family and a single training configuration. Several directions follow naturally. First, applying the geometric product explicitly across variables could capture lead–lag relations between series, which our spectral analysis suggests is a natural fit. Second, higher-grade products and learnable shift sets could relate TS-CAN more closely to higher-level path signatures. Third, the interaction could be scaled up in time series foundation models. More broadly, we hope this work encourages further study of the interaction primitive itself, alongside the architectural question of where interactions take place.

C Proofs of the Theoretical Results

Throughout the temporal proofs, we fix a single channel j of a single variable and drop both indices. Let s_n and c_n ($n = 1, \dots, N$) denote the state and context of that channel at patch n. For $k \geq 1$, the causal temporal shift sets $(\mathcal{T}_k \mathbf{c})_n = c_{n-k}$ and uses zero padding, $c_n = s_n = 0$ for $n \leq 0$. Under this shift, the wedge term of Eq. equation 8 reads

$$\mathcal{W}_k(\mathbf{s}, \mathbf{c})_n = s_n \, c_{n-k} - c_n \, s_{n-k}. \tag{13}$$

We view the sequence $P_n = (c_n, s_n) \in \mathbb{R}^2$ as the vertices of a planar path. Zero padding gives $P_n = 0$ for $n \leq 0$, so the path starts at the origin.

C.1 Proof of Proposition 1 (Discrete Lévy Area)

Lévy area. For a continuous path of bounded variation $\gamma = (x, y) : [0, T] \to \mathbb{R}^2$ started at $\gamma(0)$, the Lévy area is the antisymmetric part of its second-level signature (Lyons, 1998; Chevyrev & Kormilitzin, 2016):

$$A(\gamma) = \frac{1}{2} \int_0^T \Big(\big(x_t - x_0\big) \, \mathrm{d}y_t - \big(y_t - y_0\big) \, \mathrm{d}x_t \Big). \tag{14}$$

It equals the signed area enclosed between the path and the chord joining its endpoints. The area is positive for counter-clockwise motion.

Lemma 1 (Area of a linear segment)**.** *Let ℓ be the linear segment from $p = (x_p, y_p)$ to $q = (x_q, y_q)$. Then*

$$\int_\ell \big(x\,\mathrm{d}y - y\,\mathrm{d}x\big) = x_p\,y_q - y_p\,x_q = \det[\,p\;\;q\,],$$

which is twice the signed area of the triangle with vertices 0, p, and q.

Proof. Parametrize ℓ by $x_\tau = x_p + \tau\Delta x$ and $y_\tau = y_p + \tau\Delta y$ for $\tau \in [0,1]$, where $\Delta x = x_q - x_p$ and $\Delta y = y_q - y_p$. Then

$$\int_0^1 \big[(x_p+\tau\Delta x)\Delta y - (y_p+\tau\Delta y)\Delta x\big]\mathrm{d}\tau = x_p\Delta y - y_p\Delta x + \tfrac{1}{2}\big(\Delta x\Delta y - \Delta y\Delta x\big) = x_p\,y_q - y_p\,x_q.$$

The last expression is the determinant of the matrix with columns p and q, which is twice the signed area of the triangle $(0, p, q)$. □

Proof of Proposition 1. Set $k = 1$ in Eq. equation 13 and apply Lemma 1 with $x \equiv c$, $y \equiv s$, $p = P_{n-1}$, and $q = P_n$:

$$\mathcal{W}_1(\mathbf{s},\mathbf{c})_n = c_{n-1}\,s_n - s_{n-1}\,c_n = \det[\,P_{n-1}\;\;P_n\,] = \int_{P_{n-1}\to P_n} \big(c\,\mathrm{d}s - s\,\mathrm{d}c\big).$$

This proves the first claim: $\mathcal{W}_1(\mathbf{s},\mathbf{c})_n$ is twice the signed area swept by the segment $P_{n-1} \to P_n$ as seen from the origin. Now let γ be the piecewise-linear path $0 = P_0 \to P_1 \to \cdots \to P_N$. Because γ starts at the origin, Eq. equation 14 reduces to $A(\gamma) = \frac{1}{2}\int_\gamma (c\,\mathrm{d}s - s\,\mathrm{d}c)$. Summing over the N segments gives

$$A(\gamma) = \frac{1}{2}\sum_{n=1}^{N}\int_{P_{n-1}\to P_n}\big(c\,\mathrm{d}s - s\,\mathrm{d}c\big) = \frac{1}{2}\sum_{n=1}^{N}\mathcal{W}_1(\mathbf{s},\mathbf{c})_n.$$

The first segment starts at $P_0 = 0$ and contributes $\mathcal{W}_1(\mathbf{s},\mathbf{c})_1 = 0$, which is consistent with the zero padding. □

Multi-lag extension. The same argument covers larger shifts, as stated informally in Section 2.3.

Corollary 1 (Multi-lag Lévy areas)**.** *For any $k \geq 1$ and $r \in \{1,\dots,k\}$, let $\gamma^{(k,r)}$ be the piecewise-linear path $0 \to P_r \to P_{r+k} \to P_{r+2k} \to \cdots$, that is, the path subsampled with stride k and offset r. Then*

$$\frac{1}{2}\sum_{n=1}^{N}\mathcal{W}_k(\mathbf{s},\mathbf{c})_n = \sum_{r=1}^{k} A\big(\gamma^{(k,r)}\big).$$

Proof. By Eq. equation 13 and Lemma 1, $\mathcal{W}_k(\mathbf{s},\mathbf{c})_n = \det[\,P_{n-k}\;\;P_n\,]$, which is twice the signed area swept by the segment $P_{n-k} \to P_n$. Partition the indices $n \in \{1,\dots,N\}$ into residue classes $n \equiv r \pmod k$. For each class, the segments $P_{n-k} \to P_n$ are exactly the consecutive segments of $\gamma^{(k,r)}$. Its first segment starts at $P_{r-k} = 0$ because of zero padding. Applying the proof of Proposition 1 to each subsampled path and summing over r gives the claim. □

A set of shifts $\mathcal{K}$ therefore provides the oriented areas of the state–context path observed at several temporal resolutions. This is analogous to the multi-scale lead–lag features obtained from signatures of subsampled paths (Flint et al., 2016).

C.2 Cross-feature Lévy Area

Proposition 3 (Cross-feature Lévy area)**.** *Fix channels a and b, and let $P_n = (S_{n,a}, S_{n,b}) \in \mathbb{R}^2$. Suppose the context is a symmetric linear temporal filter of the state, $\mathbf{C}_n = \sum_m \kappa_m \mathbf{S}_{n-m}$ with $\kappa_m = \kappa_{-m}$. Then*

$$S_{n,b}C_{n,a} - S_{n,a}C_{n,b} = \sum_{m\geq 1}\kappa_m\Big(\det[P_{n-m}, P_n] - \det[P_n, P_{n+m}]\Big),$$

which is a κ-weighted contrast of the signed areas swept by the path (P_n) before and after n. For κ supported on $|m| \leq 1$, it equals twice the difference between the two local triangle areas $(0, P_{n-1}, P_n)$ and $(0, P_n, P_{n+1})$.

Remark 2 (Basis dependence)**.** *The full wedge product* $\mathbf{u} \wedge \mathbf{v}$ *transforms covariantly under orthogonal changes of basis. Sampling a subset of its coefficients does not preserve this property, so the channel branch is basis dependent. This is compatible with the rest of the architecture: element-wise gates, depthwise convolutions, BatchNorm, and LayerNorm all operate in a learned coordinate basis. The sampled planes are therefore spanned by learned features, and the relevant empirical question is whether this basis carries useful information. The channel-branch ablation in Table 3 tests this directly.*

Proof of Proposition 3. Substituting $C_{n,c} = \sum_m \kappa_m S_{n-m,c}$ for $c \in \{a, b\}$ gives

$$S_{n,b}C_{n,a} - S_{n,a}C_{n,b} = \sum_m \kappa_m \big(S_{n-m,a}S_{n,b} - S_{n-m,b}S_{n,a}\big) = \sum_m \kappa_m \det[P_{n-m}, P_n].$$

The $m = 0$ term vanishes. For $m < 0$, write $m = -m'$ and use $\kappa_{-m'} = \kappa_{m'}$:

$$\det[P_{n+m'}, P_n] = -\det[P_n, P_{n+m'}].$$

Combining the positive and negative lags yields the stated sum. By Lemma 1, each determinant is twice the signed area of the triangle formed by the origin and its two points. For $|m| \le 1$, the result is therefore twice the difference between the signed areas of the two local triangles adjacent to P_n. □

C.3 Proof of Proposition 2 (Quadrature Spectrum)

Assume (s_n, c_n) is jointly weakly stationary with cross-covariance $\gamma_{sc}(h) = \mathrm{Cov}(s_n, c_{n-h})$ for $h \in \mathbb{Z}$, and assume $\sum_h |\gamma_{sc}(h)| < \infty$.

Proof of Proposition 2. Step 1: expectation of the wedge term. For $n > k$, no padded entries appear in Eq. equation 13. Let μ_s and μ_c denote the means. Then

$$\mathbb{E}\big[s_n c_{n-k}\big] = \gamma_{sc}(k) + \mu_s\mu_c, \qquad \mathbb{E}\big[c_n s_{n-k}\big] = \mathbb{E}\big[s_m c_{m-(-k)}\big]\Big|_{m=n-k} = \gamma_{sc}(-k) + \mu_s\mu_c.$$

The second identity uses stationarity with $m = n - k$. Subtracting the two gives

$$\mathbb{E}\big[\mathcal{W}_k(\mathbf{s}, \mathbf{c})_n\big] = \gamma_{sc}(k) - \gamma_{sc}(-k) = 2\,\gamma^{-}_{sc}(k), \tag{15}$$

where $\gamma^{\pm}_{sc}(h) = \frac{1}{2}\big(\gamma_{sc}(h) \pm \gamma_{sc}(-h)\big)$ denote the even and odd parts of the cross-covariance. The mean terms cancel, so no zero-mean assumption is required.

Step 2: relation to the quadrature spectrum. The cross-spectral density is $f_{sc}(\omega) = \frac{1}{2\pi}\sum_{h\in\mathbb{Z}} \gamma_{sc}(h)e^{-\mathrm{i}\omega h}$ for $\omega \in [-\pi, \pi]$ (Priestley, 1981). Write $\gamma_{sc} = \gamma^{+}_{sc} + \gamma^{-}_{sc}$ and expand $e^{-\mathrm{i}\omega h} = \cos(\omega h) - \mathrm{i}\sin(\omega h)$. An even sequence summed against $\sin(\omega h)$ gives zero, and so does an odd sequence summed against $\cos(\omega h)$. Hence

$$f_{sc}(\omega) = \underbrace{\frac{1}{2\pi}\sum_h \gamma^{+}_{sc}(h)\cos(\omega h)}_{\text{co-spectrum } \mathfrak{c}_{sc}(\omega)} \; - \mathrm{i}\; \underbrace{\frac{1}{2\pi}\sum_h \gamma^{-}_{sc}(h)\sin(\omega h)}_{\text{quadrature spectrum } q_{sc}(\omega)}.$$

Since $\gamma^{-}_{sc}(0) = 0$ and $\gamma^{-}_{sc}(h)\sin(\omega h)$ is even in h,

$$q_{sc}(\omega) = \frac{1}{2\pi}\sum_{h\ge1} 2\gamma^{-}_{sc}(h)\sin(\omega h) = \frac{1}{2\pi}\sum_{k\ge1} \big(\gamma_{sc}(k) - \gamma_{sc}(-k)\big)\sin(\omega k) = \frac{1}{2\pi}\sum_{k\ge1} \mathbb{E}\big[\mathcal{W}_k\big]\sin(\omega k),$$

where the last equality uses Eq. equation 15. So $\{\mathbb{E}[\mathcal{W}_k]\}_{k\ge1}$ are the sine-series coefficients of q_{sc}. Conversely, orthogonality of $\{\sin(\omega k)\}_{k\ge1}$ on $[-\pi, \pi]$ gives

$$\mathbb{E}\big[\mathcal{W}_k(\mathbf{s}, \mathbf{c})_n\big] = 2\int_{-\pi}^{\pi} q_{sc}(\omega)\sin(\omega k)\,\mathrm{d}\omega, \tag{16}$$

so the expected wedge features and the quadrature spectrum determine each other. Other texts define the quadrature spectrum through $f_{sc} = \mathfrak{c}_{sc} + \mathrm{i}q_{sc}$; this only flips the overall sign. □

Consequences. Proposition 2 yields three properties that motivate the design of TS-CAN. *(a) Antisymmetry.* Swapping the roles of state and context flips the sign, $\mathbb{E}[\mathcal{W}_k(\mathbf{c},\mathbf{s})] = -\mathbb{E}[\mathcal{W}_k(\mathbf{s},\mathbf{c})]$. *(b) Blindness to non-directional structure.* If the two sequences have no lead–lag relation, in the sense that $\gamma_{sc}(h) = \gamma_{sc}(-h)$ for all h, then $q_{sc} \equiv 0$ and $\mathbb{E}[\mathcal{W}_k] = 0$ for every k. The wedge term responds only to directional structure. *(c) Complementarity.* The symmetrized in-phase product $\frac{1}{2}(s_n c_{n-k} + c_n s_{n-k})$ has expectation $2\gamma^+_{sc}(k) + \mu_s\mu_c$, and $2\gamma^+_{sc}(k)$ recovers the cosine coefficients of the co-spectrum. The two parts of the geometric product therefore split the cross-spectrum into its in-phase and quadrature components. The raw product $s_n c_{n-k}$ mixes both parts, and the SiLU gate used in $\mathcal{I}_k$ breaks this exact correspondence. The wedge term alone keeps an exact spectral meaning.

We close with two examples that make the directional behavior explicit.

Corollary 2 (Phase-shifted oscillations)**.** *Let $s_n = \sqrt{2}\cos(\omega n + \Phi)$ and $c_n = \sqrt{2}\cos(\omega n + \Phi - \theta)$, where $\Phi \sim \mathrm{Unif}[0, 2\pi)$, $\omega \in (0, \pi)$, and $\theta \in (-\pi, \pi]$ is the phase lag of the context. Then for $n > k$,*

$$\tfrac{1}{2}\,\mathbb{E}\big[s_n c_{n-k} + c_n s_{n-k}\big] = \cos(\omega k)\cos\theta, \qquad \mathbb{E}\big[\mathcal{W}_k(\mathbf{s},\mathbf{c})_n\big] = -2\sin(\omega k)\sin\theta.$$

Proof. Using $2\cos a\cos b = \cos(a-b) + \cos(a+b)$ and $\mathbb{E}_\Phi[\cos(2\omega n + 2\Phi + \cdot)] = 0$, we get $\gamma_{sc}(k) = \mathbb{E}[s_n c_{n-k}] = \cos(\omega k + \theta)$. Hence $\gamma^+_{sc}(k) = \cos(\omega k)\cos\theta$ and $\gamma^-_{sc}(k) = -\sin(\omega k)\sin\theta$. Substituting into Eq. equation 15 and into the symmetrized product gives the result. □

The in-phase part depends on $\cos\theta$, which is even in θ, so a context that leads and one that lags by the same amount produce the same value. The wedge part depends on $\sin\theta$, which is odd in θ. Its sign identifies which sequence leads, and it is largest in magnitude when the two are in quadrature ($\theta = \pm\pi/2$). This is the precise form of the cosine/sine argument given in Sections 1 and 2.3.

Corollary 3 (Pure delay)**.** *Let (s_n) be white noise with variance σ^2, and let $c_n = s_{n-\tau}$ for a delay $\tau \geq 1$, so that the state leads the context by τ steps. Then $\mathbb{E}[\mathcal{W}_k(\mathbf{s},\mathbf{c})_n] = -\sigma^2\,\mathbb{1}[k = \tau]$ for $n > k$. If instead $s_n = c_{n-\tau}$, the sign flips to $+\sigma^2\,\mathbb{1}[k = \tau]$.*

Proof. Here $\gamma_{sc}(h) = \mathbb{E}[s_n s_{n-h-\tau}] = \sigma^2\,\mathbb{1}[h = -\tau]$. By Eq. equation 15, $\mathbb{E}[\mathcal{W}_k] = \gamma_{sc}(k) - \gamma_{sc}(-k) = 0 - \sigma^2\,\mathbb{1}[k = \tau]$ for $k \geq 1$. The reversed case follows from the antisymmetry in property (a). □

Across the shift set $\mathcal{K}$, the wedge features thus peak at the true delay, and their sign encodes the direction of the lead–lag relation.

C.4 Proof of Remark 1 (Invariance of the Wedge Term)

Proof. The map $\mathbf{C} \mapsto \mathcal{W}_k(\mathbf{S},\mathbf{C}) = \mathbf{S} \odot \mathcal{T}_k\mathbf{C} - \mathbf{C} \odot \mathcal{T}_k\mathbf{S}$ is linear, because $\mathcal{T}_k$ is linear. Hence $\mathcal{W}_k(\mathbf{S}, \mathbf{C} - \mathbf{S}) = \mathcal{W}_k(\mathbf{S},\mathbf{C}) - \mathcal{W}_k(\mathbf{S},\mathbf{S})$. Moreover, $\mathcal{W}_k(\mathbf{S},\mathbf{S}) = \mathbf{S} \odot \mathcal{T}_k\mathbf{S} - \mathbf{S} \odot \mathcal{T}_k\mathbf{S} = \mathbf{0}$. This holds for every shift operator, whether cyclic, causal, or channel-wise. For the in-phase term, $\mathbf{S} \odot \mathcal{T}_k(\mathbf{C} - \mathbf{S}) = \mathbf{S} \odot \mathcal{T}_k\mathbf{C} - \mathbf{S} \odot \mathcal{T}_k\mathbf{S}$, so the differential context removes exactly the self-interaction $\mathbf{S} \odot \mathcal{T}_k\mathbf{S}$ before the SiLU gate. □

C.5 Surrogate Intervention and Operator Verification

IAAFT surrogates. For each variable of ETTh1, ETTh2, ETTm1, and ETTm2, we generate an iterative amplitude-adjusted Fourier transform (IAAFT) surrogate (Schreiber & Schmitz, 1996; Theiler et al., 1992). Starting from a random permutation of the series, the algorithm alternates between two steps: (i) it imposes the original Fourier amplitudes while keeping the current phases, and (ii) it rank-orders the result to restore the original amplitude distribution. The resulting surrogate has the same marginal distribution and, up to numerical precision, the same power spectrum and autocorrelation as the original series. Its Fourier phases, however, are randomized, which destroys higher-order and nonlinear temporal structure. Surrogates are generated for the entire series before the standard train/validation/test split, and variables are processed independently. Independent processing also randomizes phase relations across variables. This has little effect on TS-CAN, which processes variables independently in its backbone.

Irreversibility index. A stationary process is time-reversible if its statistics are invariant under $n \mapsto -n$. Gaussian linear processes are always reversible (Weiss, 1975). A simple indicator of irreversibility is the asymmetry of increments: for a reversible process, the increment $d_n^{(k)} = x_{n+k} - x_n$ has the same distribution as $-d_n^{(k)}$, so its skewness is zero. After standardizing each variable, we define the index as the mean absolute skewness of $d_n^{(k)}$ over lags $k \in \{1, 2, 4, 8, 16, 32\}$ and over variables. The index is not exactly zero on the surrogates because IAAFT matches the spectrum and the marginal distribution only approximately, but it decreases by 56–86% on every dataset.

Training setup. To isolate the effect of the wedge term, the surrogate intervention follows the same training protocol as the main experiments, including the optimizer, scheduler, data split, training epochs, and model hyperparameters. The in-phase-only model removes the wedge term from the temporal branch, and all other components are identical. We repeat each run with three random seeds and report the mean. Table 7 reports the absolute test MSE.

Table 7: Test MSE of the in-phase-only and full models on original and IAAFT surrogate data.

	Original		Surrogate	
Dataset	In-phase only	Full	In-phase only	Full
ETTh1	0.380	0.361	0.558	0.561
ETTh2	0.298	0.305	0.395	0.413
ETTm1	0.306	0.295	0.285	0.284
ETTm2	0.169	0.168	0.213	0.212

Operator verification. We verify Corollary 3 directly on the operator, without training. We construct a lead–lag pair from a unit-variance white-noise driver e_n and the series $x_n = e_n + ae_{n-\tau}$, with $a = 0.9$ and $\tau = 16$. Both series are standardized, and we compute the temporal wedge between them at patch resolution with stride $R = 4$. For standardized series, the corollary predicts a single non-zero response at $k = \tau/R = 4$, with value $\mp a/\sqrt{1+a^2} \approx \mp 0.669$. Table 8 confirms both the location and the sign pattern.

Table 8: Signed wedge response of the operator at temporal shift k (patch units).

Setting	$k \neq 4$	$k = 4$	Theory ($k = 4$)
$+a$	0.000	-0.672	-0.669
$-a$	0.000	$+0.669$	$+0.669$
time-reversed $+a$	0.000	$+0.672$	$+0.669$

C.6 Dataset Details

This appendix summarizes the datasets used for the complete forecasting evaluation and clarifies the baseline families included in the comparisons. For long-term forecasting we use eight widely used benchmarks: ETTh1, ETTh2, ETTm1, and ETTm2 (Zhou et al., 2021), together with Weather, Electricity (ECL), and Traffic (Wu et al., 2021), and Exchange (Lai et al., 2018). They cover hourly to 10-minute sampling rates, 7 to 862 variables, and both strongly periodic and highly non-stationary dynamics. For short-term forecasting we use the M4 competition data (Makridakis et al., 2020), which contains 100,000 univariate series at yearly to hourly frequencies, and the PEMS traffic-flow datasets (Liu et al., 2022a).

Baselines. We compare with twelve recent forecasters grouped by architecture family. MLP-based: TimeMixer++ (Wang et al., 2025a), TimeMixer (Wang et al., 2024), TiDE (Das et al., 2023), and DLinear (Zeng et al., 2023). Transformer-based: iTransformer (Liu et al., 2024), PatchTST (Nie et al., 2023), Crossformer (Zhang & Yan, 2023), FEDformer (Zhou et al., 2022), Non-stationary Transformer (Liu et al., 2022b), and Autoformer (Wu et al., 2021). Convolutional: TimesNet (Wu et al., 2023) and SCINet (Liu et al., 2022a). For M4 we additionally include N-HiTS (Challu et al., 2023) and N-BEATS (Oreshkin et al., 2020).

C.7 Implementation Details

Forecasting protocol. We follow the Time-Series-Library protocol (Wu et al., 2023; Wang et al., 2026). Long-term forecasting uses horizons $F \in \{96, 192, 336, 720\}$ and reports MSE and MAE. M4 reports SMAPE, MASE, and OWA. PEMS uses a 12-step horizon and reports MAE, MAPE, and RMSE. Lower is better for all metrics. Baseline results are taken from Wang et al. (2025a), which evaluates all models under a unified setting.

TS-CAN implementation. We implement TS-CAN in PyTorch based on the Time-Series-Library forecasting pipeline. Unless otherwise specified, all long-term forecasting experiments use MSE loss and the Adam optimizer with mixed-precision training. For the ETTh1-96 ablation study, we use input length 96, prediction length 96, two TS-CAN blocks, hidden dimension 128, FFN dimension 192, patch length 16, stride 8, batch size 8, dropout 0.05, stochastic depth 0.05, and learning rate 5×10^{-4} with cosine decay. The shift set is $\{1, 2, 4, 8, 16\}$, and both temporal and feature-dimensional geometric interactions use the same shift set unless stated otherwise. All ablation variants change only the factor under study while keeping the backbone, optimizer, data split, and training epochs fixed. Each ablation result is averaged over three random seeds.

Efficiency benchmark setup. All efficiency results in Tables 6 and 9 are measured under a single protocol so that the comparison reflects architectural cost rather than configuration differences. All models use the same lightweight width, $d_{\text{model}} = 128$ and $d_{\text{ff}} = 256$ (TS-CAN contains no FFN, so d_{ff} does not apply to it), with lookback length 96, horizon 96, and batch size 16. Every model receives the same fixed random input batch. For training cost, we run 10 warm-up iterations of forward pass, MSE loss, backward pass, and Adam update, and then average the time of the next 50 iterations. Peak GPU memory is recorded over the same 50 iterations. Inference latency is measured the same way (10 warm-up and 50 timed forward passes) in evaluation mode without gradient tracking. We call `torch.cuda.synchronize()` before and after each timed segment so that asynchronous kernel launches are fully accounted for. All measurements are performed on a single NVIDIA GeForce RTX 4090 D GPU. The unified width isolates architectural cost but differs from the tuned configurations behind the accuracy results in Table 1. The Avg. MSE column is therefore included for reference only and is not measured at the benchmarked configuration.

Table 9: Full efficiency comparison under a unified setting (lookback 96, horizon 96, batch size 16, single RTX 4090D). FLOPs are counted per batch with PyTorch's FLOP counter, which includes matrix multiplications and convolutions but not element-wise operations such as the shifted products of TS-CAN. We therefore regard measured memory and latency as the primary efficiency metrics.

Model	Params (M)	FLOPs (G)	Train mem. (GB)	Train (ms/iter)	Infer. (ms/batch)	Avg. MSE
TS-CAN	0.950	2.124	0.017	4.098	0.318	0.310
TimeMixer	0.207	7.398	0.280	13.992	2.453	0.346
iTransformer	0.290	0.102	0.024	3.274	0.664	0.345
PatchTST	0.415	0.750	0.049	3.540	0.789	0.373
Crossformer	2.013	2.753	0.112	23.455	3.943	0.607
TiDE	0.394	0.175	0.047	15.523	3.146	0.448
TimesNet	37.514	1151.685	18.726	203.802	22.291	0.381
DLinear	0.019	0.004	0.017	0.686	0.113	0.406
SCINet	0.049	0.018	0.019	31.044	5.312	0.609
FEDformer	0.865	1.709	0.136	130.559	27.117	0.418
Stationary	0.521	1.709	0.166	10.785	2.863	0.418
Autoformer	0.471	1.709	0.123	14.896	6.144	0.441

D Hyperparameter Sensitivity

We study the sensitivity of TS-CAN to its main architectural hyperparameters on ETTh1. We vary one hyperparameter or one pair at a time and keep all others at their default values ($\mathcal{K} = \{1, 2, 4, 8, 16\}$, $d = 128$, $B = 2$, $P = 16$, $R = 8$). Each configuration is trained once. All remaining hyperparameters, such as the learning rate and dropout, were tuned for the default configuration. Deviations from the default are therefore evaluated under a slight disadvantage, and part of the non-

monotonic variation below likely reflects run-to-run variance. We thus focus on the overall range of performance rather than on individual differences.

Shift set. Table 10 varies the shift set shared by the channel and temporal branches. The two configurations with five shifts perform best and are nearly identical. One uses exponential spacing ($\{1, 2, 4, 8, 16\}$) and the other linear spacing ($\{1, 2, 3, 4, 5\}$), with MSE 0.3606 and 0.3612. Configurations with one to four shifts are 3.4–6.0% worse, without a monotonic trend among them. Accuracy therefore depends mainly on the *number* of sampled bivector planes rather than on their spacing. A plausible explanation is that the pointwise projection $\mathcal{P}$ already mixes all channels, so the reach of individual shifts matters less than the diversity of sampled interactions. Since the parameters of $\mathcal{P}$ scale linearly with $|\mathcal{K}|$, smaller shift sets offer a lighter model at a moderate accuracy cost.

Table 10: Sensitivity to the shift set $\mathcal{K}$ on ETTh1 ($F = 96$). Δ is the relative change in MSE with respect to the default.

$\mathcal{K}$	$\lvert\mathcal{K}\rvert$	MSE	MAE	Δ
$\{1\}$	1	0.3728	0.3963	+3.4%
$\{1, 2\}$	2	0.3775	0.3983	+4.7%
$\{1, 2, 4\}$	3	0.3764	0.3991	+4.4%
$\{1, 2, 4, 8\}$	4	0.3823	0.4016	+6.0%
$\{1, 2, 4, 8, 16\}$ (default)	5	**0.3606**	**0.3930**	–
$\{1, 2, 3, 4, 5\}$ (linear)	5	0.3612	0.3931	+0.2%

Width and depth. Table 11 varies the hidden width d and the number of blocks B. The default ($d = 128$, $B = 2$) performs best. Of the other 15 configurations, 11 are within 5% of it, and all are within 9%. Two observations stand out. First, a single block is not enough at $d = 128$ (+8.1%). With two or more blocks, the MSE stays within 2% of the default, which indicates that stacking geometric interactions is beneficial. Second, much smaller models remain competitive: $d = 32$ with $B = 4$ and $d = 64$ with $B = 3$ are only 3.7% and 3.0% worse, respectively, while using far fewer parameters, since the projections scale as d^2. Increasing the width to $d = 256$ brings no improvement, and at $B = 3$ it gives the worst result in the grid. This is consistent with overfitting on the relatively small ETTh1 training set. Overall, TS-CAN is not highly sensitive to capacity, and the no-FFN design scales down gracefully.

Table 11: Sensitivity to width d and depth B on ETTh1 ($F = 96$), MSE. The default configuration is highlighted.

d	$B = 1$	$B = 2$	$B = 3$	$B = 4$
32	0.3781	0.3763	0.3817	0.3738
64	0.3731	0.3772	0.3714	0.3864
128	0.3897	**0.3606**	0.3676	0.3654
256	0.3748	0.3742	0.3923	0.3676

Patch length. Table 12 varies the patch length P, with the stride set to $R = P/2$. The default $P = 16$ performs best and $P = 24$ is second (+2.1%). The shortest and longest patches are the least accurate: $P = 8$ is 2.9% worse, with the highest MAE, and $P = 32$ is 5.6% worse. Two effects are at play. Very short patches carry little local structure per token, which makes each in-phase and wedge interaction noisier. Very long patches coarsen the temporal resolution of the wedge. Overall, moderate patch lengths ($P \in [16, 24]$) work best.

D.1 Qualitative Forecasting Showcases

Figures 2–4 provide qualitative examples of the forecasting behavior behind the aggregate results. Across the ETT datasets, TS-CAN tracks trend changes and local oscillations without collapsing to an overly smooth forecast. On ECL and Traffic, where the target channels contain sharper bursts and sensor-specific fluctuations, TS-CAN preserves more local variation while keeping stable intervals



Figure 2: Qualitative forecasting examples on ETTh1 (top) and ETTh2 (bottom). The dashed vertical line separates the lookback window and the prediction window.

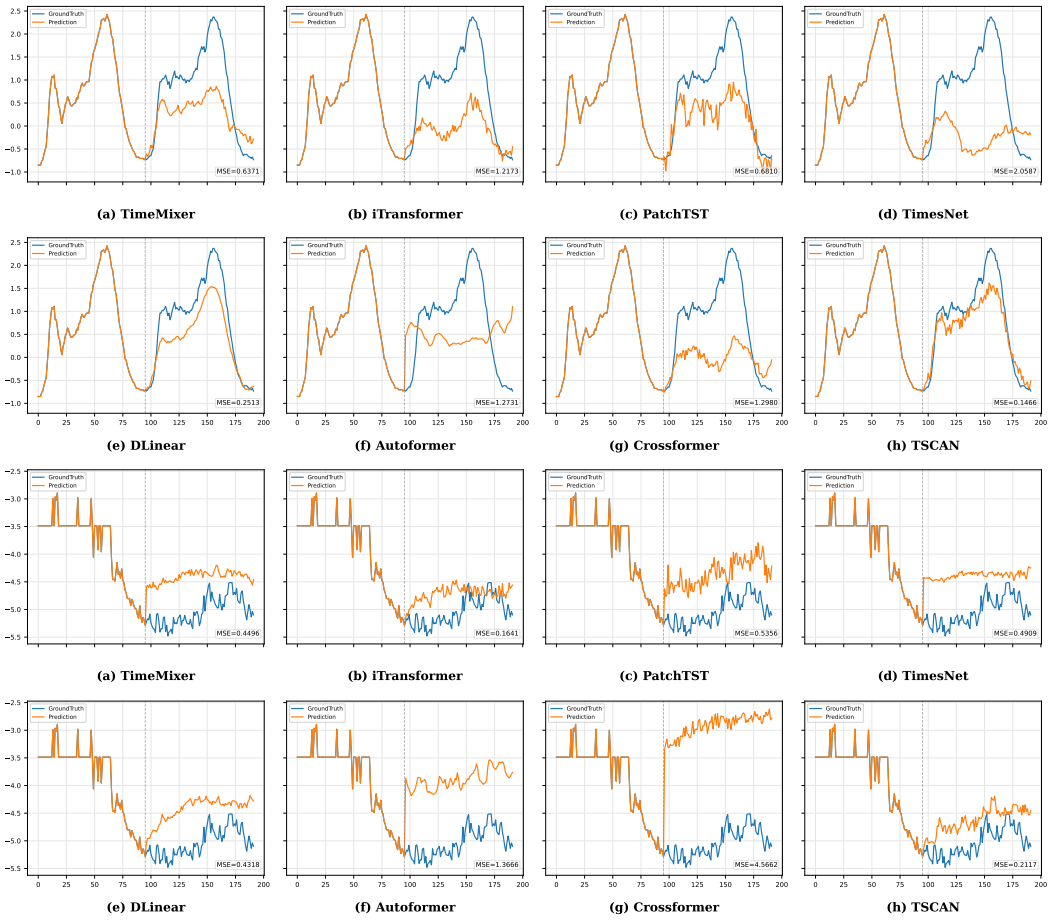


Figure 3: Qualitative forecasting examples on ETTm1 (top) and ETTm2 (bottom). The dashed vertical line separates the lookback window and the prediction window.

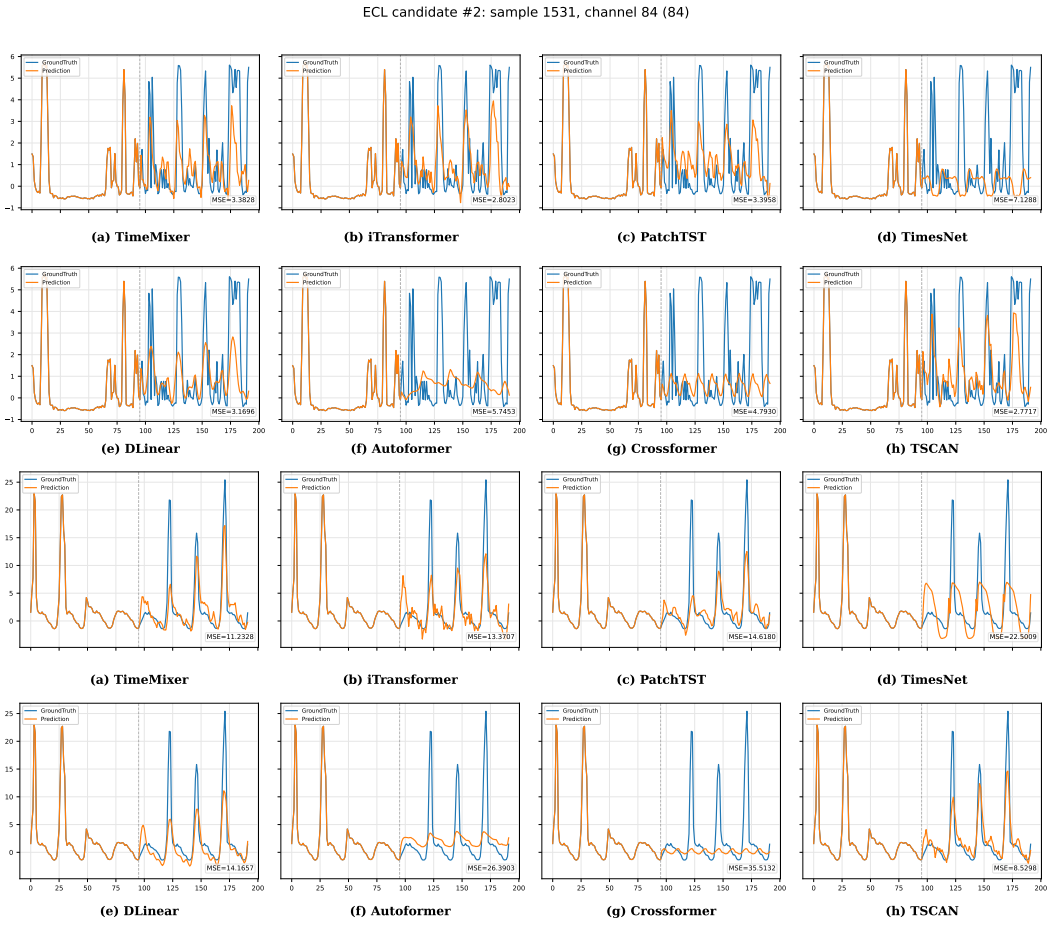


Figure 4: Qualitative forecasting examples on Electricity (ECL; top) and Traffic (bottom). The dashed vertical line separates the lookback window and the prediction window.

Table 12: Sensitivity to patch length P (stride $R = P/2$) on ETTh1 ($F = 96$).

P / R	MSE	MAE	Δ
8 / 4	0.3710	0.4046	+2.9%
12 / 6	0.3763	0.4013	+4.4%
16 / 8 (default)	**0.3606**	**0.3930**	–
24 / 12	0.3683	0.3945	+2.1%
32 / 16	0.3808	0.3985	+5.6%

well aligned. These cases suggest that the geometric interaction is useful not only for smooth trends, but also for local directional changes and abrupt fluctuations.

E Full Short-Term Results

Tables 13 and 14 provide the complete short-term forecasting results. M4 contains four frequency subsets, while PEMS evaluates 12-step traffic-flow forecasting on multiple sensor networks.

F Full Long-Term Results

This appendix provides the horizon-wise long-term forecasting results behind the averaged main-table comparison. For each dataset, *Avg* averages the four prediction horizons $\{96, 192, 336, 720\}$.

Table 13: Full short-term forecasting results on M4. All prediction lengths are in $[6, 48]$. Lower SMAPE, MASE, and OWA indicate better prediction. **Red**: best; Blue: second best.

Dataset	Metric	**TS-CAN**	TimeMixer++	TimeMixer	iTransformer	TiDE	TimesNet	N-HiTS	N-BEATS*	PatchTST	MICN	FiLM	LightTS	DLinear	FEDformer	Stationary	Autoformer
Yearly	SMAPE	**13.132**	13.179	13.206	13.923	15.320	13.387	13.418	13.436	16.463	25.022	17.431	14.247	16.965	13.728	13.717	13.974
	MASE	2.932	2.934	**2.916**	3.214	3.540	2.996	3.045	3.043	3.967	7.162	4.043	3.109	4.283	3.048	3.078	3.134
	OWA	0.771	**0.769**	0.776	0.830	0.910	0.786	0.793	0.794	1.003	1.667	1.042	0.827	1.058	0.803	0.807	0.822
Quarterly	SMAPE	9.890	**9.755**	9.996	10.757	11.830	10.100	10.202	10.124	10.644	15.214	12.925	11.364	12.145	10.792	10.958	11.338
	MASE	**1.141**	1.159	1.166	1.283	1.410	1.182	1.194	1.169	1.278	1.963	1.664	1.328	1.520	1.283	1.325	1.365
	OWA	0.865	0.865	**0.825**	0.956	1.050	0.890	0.899	0.886	0.949	1.407	1.193	1.000	1.106	0.958	0.981	1.012
Monthly	SMAPE	**12.338**	12.432	12.605	13.796	15.180	12.670	12.791	12.677	13.399	16.943	15.407	14.014	13.514	14.260	13.917	13.958
	MASE	**0.901**	0.904	0.919	1.083	1.190	0.933	0.969	0.937	1.031	1.442	1.298	1.053	1.037	1.102	1.097	1.103
	OWA	0.851	**0.841**	0.869	0.987	1.090	0.878	0.899	0.880	0.949	1.265	1.144	0.981	0.956	1.012	0.998	1.002
Others	SMAPE	**4.501**	4.698	4.564	5.569	6.120	4.891	5.061	4.925	6.558	41.985	7.134	15.880	6.709	4.954	6.302	5.485
	MASE	3.019	**2.931**	3.115	3.940	4.330	3.302	3.216	3.391	4.511	62.734	5.090	11.434	4.953	3.264	4.064	3.865
	OWA	**0.950**	1.010	0.982	1.207	1.330	1.035	1.040	1.053	1.401	14.313	1.553	3.474	1.487	1.036	1.304	1.187

* Following the TimeMixer++ comparison, N-BEATS is evaluated without its ensemble.

Table 14: Full short-term forecasting results on PEMS. All input lengths are 96 and prediction lengths are 12. Lower MAE, MAPE, and RMSE indicate better prediction. **Red**: best; Blue: second best.

Dataset	Metric	**TS-CAN**	TimeMixer++	TimeMixer	iTransformer	TiDE	SCINet	Crossformer	PatchTST	TimesNet	MICN	DLinear	FEDformer	Stationary	Autoformer
PEMS03	MAE	14.476	**13.990**	14.630	16.720	18.390	15.970	15.640	18.950	16.410	15.710	19.700	19.000	17.640	18.080
	MAPE	13.397	13.430	**11.540**	15.810	17.390	15.890	15.740	17.290	15.170	15.670	18.350	18.570	17.560	18.750
	RMSE	23.292	24.030	**23.280**	27.810	30.590	25.200	25.560	30.150	26.720	24.550	32.350	30.050	28.370	27.820
PEMS04	MAE	18.383	**17.460**	19.210	21.810	23.990	20.350	20.380	24.860	21.630	21.620	24.620	26.510	22.340	25.000
	MAPE	12.036	**11.340**	12.530	13.420	14.760	12.840	12.840	16.650	13.150	13.530	16.120	16.760	14.850	16.700
	RMSE	30.133	**28.830**	30.920	33.910	37.300	32.310	32.410	40.460	34.900	34.390	39.510	41.810	35.470	38.020
PEMS07	MAE	19.909	**18.380**	20.570	23.010	25.310	22.790	22.540	27.870	25.120	22.280	28.650	27.920	26.020	26.920
	MAPE	8.661	**7.320**	8.620	10.020	11.020	9.410	9.380	12.690	10.600	9.570	12.150	12.290	11.750	11.830
	RMSE	32.552	**31.750**	33.590	35.560	39.120	35.610	35.490	42.560	40.710	35.400	45.020	42.290	42.340	40.600
PEMS08	MAE	**13.721**	13.810	15.220	17.940	19.730	17.380	17.560	20.350	19.010	17.760	20.260	20.560	19.290	20.470
	MAPE	8.945	**8.210**	9.670	10.930	12.020	10.800	10.920	13.150	11.830	10.760	12.090	12.410	12.210	12.270
	RMSE	**23.011**	23.620	24.260	27.880	30.670	27.340	27.210	31.040	30.650	27.260	32.380	32.970	38.620	31.520

Table 15: Full results for the long-term forecasting task. We compare competitive models under different prediction lengths. *Avg* is averaged over prediction lengths $\{96, 192, 336, 720\}$.

Models		**TS-CAN**		TimeMixer++		TimeMixer		iTransformer		PatchTST		Crossformer		TiDE		TimesNet		DLinear		SCINet		FEDformer		Stationary		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.151**	**0.198**	0.155	0.205	0.163	0.209	0.174	0.214	0.186	0.227	0.195	0.271	0.202	0.261	0.172	0.220	0.195	0.252	0.221	0.306	0.217	0.296	0.173	0.223	0.266	0.336
	192	**0.196**	**0.241**	0.201	0.245	0.208	0.250	0.221	0.254	0.234	0.265	0.209	0.277	0.242	0.298	0.219	0.261	0.237	0.295	0.261	0.340	0.276	0.336	0.245	0.285	0.307	0.367
	336	**0.237**	0.275	**0.237**	**0.265**	0.251	0.287	0.278	0.296	0.284	0.301	0.273	0.332	0.287	0.335	0.280	0.306	0.282	0.331	0.309	0.378	0.339	0.380	0.321	0.338	0.359	0.395
	720	**0.311**	**0.328**	0.312	0.334	0.339	0.341	0.358	0.347	0.356	0.349	0.379	0.401	0.351	0.386	0.365	0.359	0.345	0.382	0.377	0.427	0.403	0.428	0.414	0.410	0.419	0.428
	Avg	**0.224**	**0.261**	0.226	0.262	0.240	0.271	0.258	0.278	0.265	0.285	0.264	0.320	0.271	0.320	0.259	0.287	0.265	0.315	0.292	0.363	0.309	0.360	0.288	0.314	0.338	0.382
Electricity	96	**0.135**	0.231	**0.135**	**0.222**	0.153	0.247	0.148	0.240	0.190	0.296	0.219	0.314	0.237	0.329	0.168	0.272	0.210	0.302	0.247	0.345	0.193	0.308	0.169	0.273	0.201	0.317
	192	**0.147**	0.243	**0.147**	**0.235**	0.166	0.256	0.162	0.253	0.199	0.304	0.231	0.322	0.236	0.330	0.184	0.322	0.210	0.305	0.257	0.355	0.201	0.315	0.182	0.286	0.222	0.334
	336	**0.164**	0.266	**0.164**	**0.245**	0.185	0.277	0.178	0.269	0.217	0.319	0.246	0.337	0.249	0.344	0.198	0.300	0.223	0.319	0.269	0.369	0.214	0.329	0.200	0.304	0.231	0.443
	720	**0.208**	**0.296**	0.212	0.310	0.225	0.310	0.225	0.317	0.258	0.352	0.280	0.363	0.284	0.373	0.220	0.320	0.258	0.350	0.299	0.390	0.246	0.355	0.222	0.321	0.254	0.361
	Avg	**0.164**	0.259	0.165	**0.253**	0.182	0.272	0.178	0.270	0.216	0.318	0.244	0.334	0.251	0.344	0.192	0.304	0.225	0.319	0.268	0.365	0.214	0.327	0.193	0.296	0.227	0.338
Traffic	96	**0.361**	0.261	0.392	**0.253**	0.462	0.285	0.395	0.268	0.526	0.347	0.644	0.429	0.805	0.493	0.593	0.321	0.650	0.396	0.788	0.499	0.587	0.366	0.612	0.338	0.613	0.388
	192	**0.370**	0.271	0.402	**0.258**	0.473	0.296	0.417	0.276	0.522	0.332	0.665	0.431	0.756	0.474	0.617	0.336	0.598	0.370	0.789	0.505	0.604	0.373	0.613	0.340	0.616	0.382
	336	**0.388**	0.279	0.428	**0.263**	0.498	0.296	0.433	0.283	0.517	0.334	0.674	0.420	0.762	0.477	0.629	0.336	0.605	0.373	0.797	0.508	0.621	0.383	0.618	0.328	0.622	0.337
	720	**0.427**	0.292	0.441	**0.282**	0.506	0.313	0.467	0.302	0.552	0.352	0.683	0.424	0.719	0.449	0.640	0.350	0.645	0.394	0.841	0.523	0.626	0.382	0.653	0.355	0.660	0.408
	Avg	**0.387**	0.276	0.416	**0.264**	0.484	0.297	0.428	0.282	0.529	0.341	0.667	0.426	0.760	0.473	0.620	0.336	0.625	0.383	0.804	0.509	0.610	0.376	0.624	0.340	0.628	0.379
Exchange	96	**0.082**	**0.203**	0.085	0.214	0.090	0.235	0.086	0.206	0.088	0.205	0.256	0.367	0.094	0.218	0.107	0.234	0.088	0.218	0.267	0.396	0.148	0.278	0.111	0.237	0.197	0.323
	192	**0.172**	**0.296**	0.175	0.313	0.187	0.343	0.177	0.299	0.176	0.299	0.470	0.509	0.184	0.307	0.226	0.344	0.176	0.315	0.351	0.459	0.271	0.315	0.219	0.335	0.300	0.369
	336	0.307	0.403	0.316	0.420	0.353	0.473	0.331	0.417	**0.301**	**0.397**	1.268	0.883	0.349	0.431	0.367	0.448	0.313	0.427	1.324	0.853	0.460	0.427	0.421	0.476	0.509	0.524
	720	**0.792**	**0.666**	0.851	0.689	0.934	0.761	0.847	0.691	0.901	0.714	1.767	1.068	0.852	0.698	0.964	0.746	0.839	0.695	1.058	0.797	1.195	0.695	1.092	0.769	1.447	0.941
	Avg	**0.338**	0.392	0.357	**0.391**	0.391	0.453	0.360	0.403	0.367	0.404	0.940	0.707	0.370	0.413	0.416	0.443	0.354	0.414	0.750	0.626	0.519	0.429	0.461	0.454	0.613	0.539
ETTh1	96	**0.361**	**0.393**	**0.361**	0.403	0.375	0.400	0.386	0.405	0.460	0.447	0.423	0.448	0.479	0.464	0.384	0.402	0.397	0.412	0.654	0.599	0.395	0.424	0.513	0.491	0.449	0.459
	192	**0.409**	0.425	0.416	0.441	0.429	**0.421**	0.441	0.512	0.477	0.429	0.471	0.474	0.525	0.492	0.436	0.429	0.446	0.441	0.719	0.631	0.469	0.470	0.534	0.504	0.500	0.482
	336	**0.429**	**0.434**	0.430	**0.434**	0.484	0.458	0.487	0.458	0.546	0.496	0.570	0.546	0.565	0.515	0.491	0.469	0.489	0.467	0.778	0.659	0.530	0.499	0.588	0.535	0.521	0.496
	720	**0.449**	0.467	0.467	**0.451**	0.498	0.482	0.503	0.491	0.544	0.517	0.653	0.621	0.594	0.558	0.521	0.500	0.513	0.510	0.836	0.699	0.598	0.544	0.643	0.616	0.514	0.512
	Avg	**0.412**	**0.430**	0.419	0.432	0.447	0.440	0.454	0.447	0.516	0.484	0.529	0.522	0.541	0.507	0.458	0.450	0.461	0.457	0.747	0.647	0.498	0.484	0.570	0.537	0.496	0.487
ETTh2	96	0.280	0.334	**0.276**	**0.328**	0.289	0.341	0.297	0.349	0.308	0.355	0.745	0.584	0.400	0.440	0.340	0.374	0.340	0.394	0.707	0.621	0.358	0.397	0.476	0.458	0.346	0.388
	192	0.345	0.383	**0.342**	**0.379**	0.372	0.392	0.380	0.400	0.393	0.405	0.877	0.656	0.528	0.509	0.402	0.414	0.482	0.479	0.860	0.689	0.429	0.439	0.512	0.493	0.456	0.452
	336	0.347	**0.393**	**0.346**	0.398	0.386	0.414	0.428	0.432	0.427	0.436	1.043	0.731	0.643	0.571	0.452	0.452	0.591	0.541	1.000	0.744	0.496	0.487	0.552	0.551	0.482	0.486
	720	**0.392**	0.426	**0.392**	**0.415**	0.412	0.434	0.427	0.445	0.436	0.450	1.104	0.763	0.874	0.679	0.462	0.468	0.839	0.661	1.249	0.838	0.463	0.474	0.562	0.560	0.515	0.511
	Avg	0.341	0.384	**0.339**	**0.380**	0.364	0.395	0.383	0.407	0.391	0.411	0.942	0.684	0.611	0.550	0.414	0.427	0.563	0.519	0.954	0.723	0.437	0.449	0.526	0.516	0.450	0.459
ETTm1	96	**0.286**	0.345	0.310	**0.334**	0.320	0.357	0.334	0.368	0.352	0.374	0.404	0.426	0.364	0.387	0.338	0.375	0.346	0.374	0.418	0.438	0.379	0.419	0.386	0.398	0.505	0.475
	192	**0.332**	0.370	0.348	**0.362**	0.361	0.381	0.390	0.393	0.374	0.387	0.450	0.451	0.398	0.404	0.374	0.387	0.382	0.391	0.439	0.450	0.426	0.441	0.459	0.444	0.553	0.496
	336	**0.369**	0.395	0.376	**0.391**	0.390	0.404	0.426	0.420	0.421	0.414	0.532	0.515	0.428	0.425	0.410	0.411	0.415	0.415	0.490	0.485	0.445	0.459	0.495	0.464	0.621	0.537
	720	**0.422**	0.426	0.440	**0.423**	0.454	0.441	0.491	0.459	0.462	0.449	0.666	0.589	0.487	0.461	0.478	0.450	0.473	0.451	0.595	0.550	0.543	0.490	0.585	0.516	0.671	0.561
	Avg	**0.352**	0.384	0.369	**0.378**	0.381	0.395	0.407	0.410	0.406	0.407	0.513	0.495	0.419	0.419	0.400	0.406	0.404	0.408	0.485	0.481	0.448	0.452	0.481	0.456	0.588	0.517
ETTm2	96	**0.168**	0.258	0.170	**0.245**	0.175	0.258	0.180	0.264	0.183	0.270	0.287	0.366	0.207	0.305	0.187	0.267	0.193	0.293	0.286	0.377	0.203	0.287	0.192	0.274	0.255	0.339
	192	**0.228**	0.298	0.229	**0.291**	0.237	0.299	0.250	0.309	0.255	0.314	0.414	0.492	0.290	0.364	0.249	0.309	0.284	0.361	0.399	0.445	0.269	0.328	0.280	0.339	0.281	0.340
	336	**0.280**	**0.330**	0.303	0.343	0.298	0.340	0.311	0.348	0.309	0.347	0.597	0.542	0.377	0.422	0.321	0.351	0.382	0.429	0.637	0.591	0.325	0.366	0.334	0.361	0.339	0.372
	720	**0.357**	**0.388**	0.373	0.399	0.391	0.396	0.412	0.407	0.412	0.404	1.730	1.042	0.558	0.524	0.408	0.403	0.558	0.525	0.960	0.735	0.421	0.415	0.417	0.413	0.433	0.432
	Avg	**0.258**	**0.319**	0.269	0.320	0.275	0.323	0.288	0.332	0.290	0.334	0.757	0.610	0.358	0.404	0.291	0.333	0.354	0.402	0.954	0.723	0.305	0.349	0.306	0.347	0.327	0.371