SIMPLE TEMPERATURE SCALING OF CHANNEL GRAPHS IS ENOUGH FOR MULTIVARIATE TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

In multivariate time series forecasting, modeling channels independently can miss useful dependencies, whereas jointly modeling them may introduce irrelevant or noisy interactions, with the appropriate interaction scope varying across datasets. How to continuously control and theoretically characterize this scope within a unified framework spanning Channel Independence (CI), Channel Partiality (CP), and Channel Dependence (CD) remains an open question. We view CI, CP, and CD as different concentration regimes of a common channel interaction distribution. Building on this view, we propose τformer, a temporal Transformer that continuously regulates interaction scope through temperature scaling of a learned channel graph. Learned channel embeddings determine relative channel preferences, while temperature controls how strongly these preferences are expressed in the propagation weights. We derive temperature-dependent propagation bounds and introduce two complementary graph metrics to characterize the resulting channel interaction patterns. Extensive experiments on nine benchmark datasets across four prediction horizons demonstrate the competitive forecasting performance of τformer. Further analyses validate its ability to accommodate different channel interaction requirements and its robustness under input corruption. Our code is available at `https://anonymous.4open.science/r/Tauformer-2F29/`.

1 INTRODUCTION

Multivariate time series forecasting (MTSF) predicts the future trajectories of multiple variables from their joint historical observations, with applications in meteorology (Lam et al., 2023; Han et al., 2025), energy (Fu & Nguyen, 2003), and transportation (Qu et al., 2019). In long-horizon settings, accurate forecasting requires modeling two complementary structures: temporal dynamics within each variable and predictive relations across variables. Recent advances have substantially improved temporal modeling through attention (Zhou et al., 2021), decomposition (Wu et al., 2021), patching (Nie et al., 2022), and multi-scale representations (Wang et al., 2024a). Yet a fundamental question in channel modeling remains: *how broadly should each variable interact with others?*

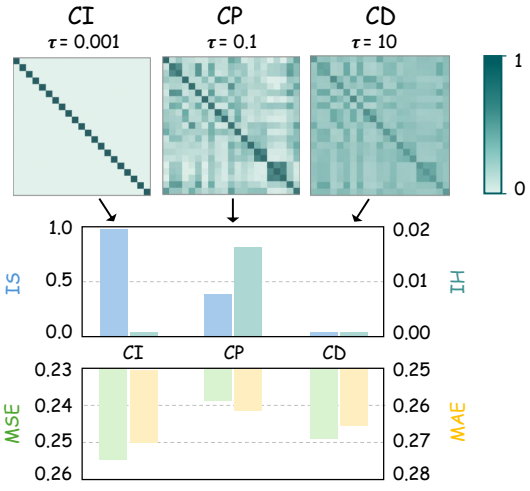

Figure 1: Weather averages over 96, 192, 336, and 720 steps (seed 999). Matrices average five layers per model and four horizons, sharing a nonlinear color scale. Bars show IS/IH; mean MSE/MAE extend downward from 0.230/0.250.

Existing approaches offer three prevailing answers (Qiu et al., 2025a). Channel Independence (CI) models each variable independently (Nie et al., 2022), avoiding cross-channel interference but potentially missing useful dependencies; Channel Dependence (CD) jointly models channels to exploit shared information (Liu et al.,

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.

2024), but may introduce irrelevant or noisy interactions; Channel Partiality (CP) selectively retains informative dependencies, seeking a balance between the two. However, many existing methods hard-code the interaction scope into specific architectures or discrete selection rules (Wu et al., 2020; Qiu et al., 2025b; Hu et al., 2025; Beichter et al., 2025; Benechehab et al., 2025; Qin et al., 2025), treating isolated, selective, and broad interaction as separate modeling choices. PCD enables continuous masking, but provides limited control and theoretical characterization of the resulting interaction scope (Lee et al., 2026). Since the appropriate interaction scope varies across multivariate systems, a fundamental question remains: *How can channel interaction scope be continuously controlled and theoretically characterized within a unified framework spanning CI, CP, and CD?*

Our key insight is that CI, CP, and CD can be viewed as different concentration regimes of a common channel interaction distribution. In a channel graph, CI concentrates interaction weights on the channel itself, CP emphasizes a small set of stronger relations, and CD distributes weights more broadly. Given learned channel similarities, temperature τ provides a simple, continuous control over this concentration: lower temperatures sharpen interaction preferences, whereas higher temperatures flatten them toward broader mixing. We further use *Identity Strength* (IS) and *Interaction Heterogeneity* (IH) to summarize self-focus and relational heterogeneity, respectively. Figure 1 illustrates this continuum on the Weather dataset: increasing τ transforms the learned graph from a CI-like diagonal structure, through selective CP-like interactions, to a diffuse CD-like regime, with the intermediate regime yielding the lowest forecasting error in this example.

Building on this insight, we propose τformer, a temporal Transformer that regulates channel interaction through a temperature-controlled graph. Each encoder layer learns channel embeddings and converts their pairwise similarities into row-normalized propagation weights through temperature scaling. Rather than directly mixing channel-specific temporal content, τformer propagates only query representations across the graph before temporal self-attention over patches, allowing cross-channel relations to guide which historical patterns are retrieved while keeping keys and values channel-specific. This design separates *whom to interact with*, determined by learned channel relations, from *how broadly to interact*, controlled by temperature, enabling different interaction scopes within a single architecture. Our contributions are threefold:

- We present a unified view of CI, CP, and CD as different concentration regimes of channel interaction, and use temperature scaling to continuously regulate the interaction scope. We further introduce Identity Strength and Interaction Heterogeneity to characterize the resulting channel graphs.
- We propose τformer, which integrates learned channel relations into temporal attention through query-only graph propagation, allowing cross-channel context to guide temporal pattern retrieval without directly mixing channel-specific temporal content.
- Across nine multivariate forecasting benchmarks and four prediction horizons, τformer achieves strong forecasting performance. Further analyses show that it captures distinct interaction regimes across datasets, distinguishes stronger from weaker channel relations, and remains robust under input corruption.

2 Related Works

2.1 Temporal Modeling in Multivariate Forecasting

Temporal modeling methods use different representations of historical patterns (Liu et al., 2025; Wang et al., 2025b; Garza et al., 2023; Shi et al., 2024; Wang et al.; Yuan et al., 2025; Gao et al., 2026). Informer reduces attention costs for long sequences with ProbSparse attention (Zhou et al., 2021), while DLinear shows that linear models with decomposition can rival Transformers (Zeng et al., 2023). PatchTST preserves local structure and extends temporal context through patch tokens (Nie et al., 2022), and Pathformer adaptively selects pathways across patch scales (Chen et al., 2024b). TimesNet represents multiple periods in 2D (Wu et al., 2023); TimeMixer mixes decomposed patterns across sampling scales with MLPs (Wang et al., 2024a); and TimeMixer++ models multiple time scales and frequency resolutions (Wang et al., 2025a). FiLM combines Legendre projections for historical information with Fourier processing to reduce noise (Zhou et al., 2022). SSformer combines sparse-scale modeling with bidirectional awareness (Liu et al., 2026). TimeXer combines patch and variate attention for endogenous and exogenous variables (Wang et al., 2024b).

2.2 Channel Modeling Strategies

Channel strategies differ in interaction scope, commonly described as CI, CD, and CP (Qiu et al., 2025a; Xiao et al., 2025; Ekambaram et al., 2024; Faw et al., 2025; Goswami et al., 2024; Das et al., 2024; Cheng et al., 2025; Feofanov et al., 2025; Shi et al., 2025). CI processes channels separately, as in DLinear and PatchTST (Zeng et al., 2023; Nie et al., 2022). CD enables broad interaction through variate attention in iTransformer (Liu et al., 2024) or a global series core in SOFTS (Han et al., 2024). CP selectively combines channel information. MTGNN uses directed top-K neighborhoods (Wu et al., 2020); LIFT transfers information from leading to lagging variables (Zhao & Shen, 2024); and CCM dynamically groups channels by similarity (Chen et al., 2024a). DUET combines temporal clustering with soft channel clustering in the frequency domain and sparsification (Qiu et al., 2025b). TimeFilter filters graphs for individual patches (Hu et al., 2025), while xCPD uses graph decomposition to route interactions between channels by frequency response (Li et al., 2026).

Graph learning provides explicit representations of channel relations. AGCRN derives dependencies from learnable node embeddings (Bai et al., 2020), GTS jointly learns a probabilistic graph (Shang et al., 2021), and FourierGNN uses a dense spatiotemporal graph in the frequency domain (Yi et al., 2023). MSGNet combines adaptive MixHop graph convolution at different scales with temporal attention (Cai et al., 2024). PCD offers a common mask formulation of channel strategies, using learnable continuous similarity masks to modulate attention logits before softmax (Lee et al., 2026). Unlike existing methods, we combine continuous scope control and theoretical characterization in a unified graph view of CI, CP, and CD, with weight concentration regulated by temperature τ.

3 Methodology

3.1 Problem Setup and Input Representation

Given historical observations $\mathbf{X} \in \mathbb{R}^{L\times N}$ of N variables over L time steps, multivariate time series forecasting aims to predict the future sequence $\mathbf{Y} \in \mathbb{R}^{T\times N}$ over a horizon of length T. We learn a mapping $\widehat{\mathbf{Y}} = f_\theta(\mathbf{X})$ that captures both temporal dependencies within each channel and predictive relations across channels.

Each channel is normalized along time within the input window and divided into M overlapping patches of length P. Here, N denotes the number of channels and M the number of patches per channel, reversing the notation used in PatchTST (Nie et al., 2022). Each patch is mapped to the hidden dimension D through a shared linear projection (Figure 2(a)). A batch of B samples yields the representation $\mathbf{H}^{(0)} \in \mathbb{R}^{B\times N\times M\times D}$.

3.2 Temperature Scaling of Channel Graphs

Within each attention block, we construct a channel graph by applying temperature scaling to learned channel similarities (Figure 2(b)). In block ℓ, each channel i has an adaptive channel embedding $\mathbf{e}_i^{(\ell)} \in \mathbb{R}^{D_e}$, learned for graph construction (Bai et al., 2020). We define nonnegative channel similarities:

$$C_{ij}^{(\ell)} = \mathrm{ReLU}\Big(\langle \mathbf{e}_i^{(\ell)}, \mathbf{e}_j^{(\ell)}\rangle / \big(\|\mathbf{e}_i^{(\ell)}\|_2 \|\mathbf{e}_j^{(\ell)}\|_2\big)\Big). \tag{1}$$

For nonzero channel embeddings, normalization gives $C_{ii}^{(\ell)} = 1$.

Inspired by the temperature relaxation in GTS (Shang et al., 2021), we use softmax with a temperature hyperparameter $\tau > 0$ to obtain deterministic propagation weights $A_{\tau,ij}^{(\ell)}$ from channel j to channel i, without discrete edge sampling:

$$A_{\tau,ij}^{(\ell)} = \frac{\exp(C_{ij}^{(\ell)}/\tau)}{\sum_{r=1}^{N} \exp(C_{ir}^{(\ell)}/\tau)}. \tag{2}$$

The resulting matrix $\mathbf{A}_\tau^{(\ell)} \in \mathbb{R}^{N\times N}$ is row-stochastic. The adaptive embeddings determine relative channel preferences, while temperature controls how strongly these preferences are expressed in the propagation weights. Each layer learns its own embeddings and shares its graph across samples.

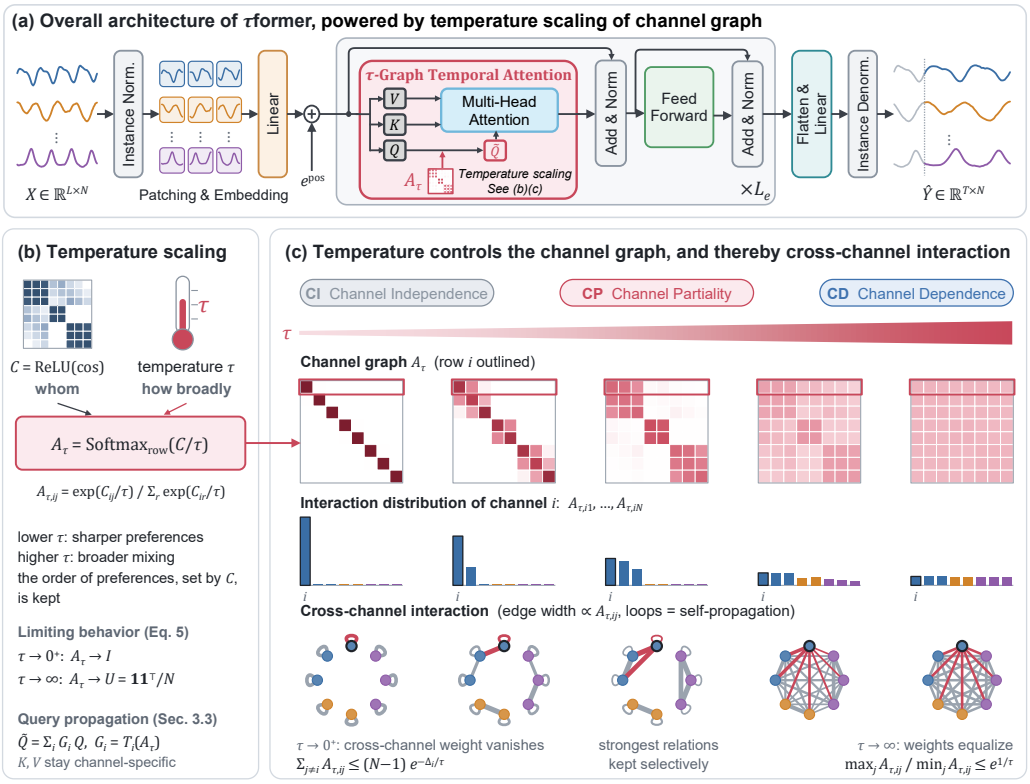

Figure 2: Framework of τformer, which controls channel interaction through temperature scaling of channel graphs. (a) Normalized series are embedded as temporal patches and passed through stacked encoder blocks to produce forecasts. (b) Within each block, similarities between learned channel embeddings determine relative channel preferences, while temperature controls the concentration of graph weights without changing their ordering. The resulting graph propagates queries across channels to guide attention over each channel's own historical keys and values. (c) For fixed similarities, increasing temperature redistributes weights in the channel graph, shifting cross-channel interaction from the CI limit through selective CP toward uniform mixing in the CD limit within the same architecture.

To isolate the effect of temperature on the propagation weights, we analyze one layer with fixed nonzero embeddings and finite $N > 1$. The layer index is omitted, and $A_{\tau,ij}$ denotes entry (i, j) of $\mathbf{A}_\tau$, equivalently written as $A_{ij}^{(\tau)}$. For any two entries in row i, the common softmax denominator cancels: $A_{\tau,ij}/A_{\tau,ir} = \exp\big((C_{ij} - C_{ir})/\tau\big)$. Decreasing τ increases the relative weight of the more similar channel, whereas increasing τ moves the ratio towards one without changing the ordering of preferences. The following theorem gives upper bounds on weight contrast and total cross-channel weight at finite temperatures.

Theorem 1 (Temperature-controlled propagation bounds)**.** *Fix the nonzero channel embeddings in one layer, with $N > 1$. The adjacency in Eq. (2) satisfies, for every $\tau > 0$ and row i,*

$$\frac{\max_j A_{\tau,ij}}{\min_j A_{\tau,ij}} \le \exp(1/\tau). \tag{3}$$

If every row has a unique diagonal maximum, let $\Delta_i = 1 - \max_{j \neq i} C_{ij} > 0$. For every $\tau > 0$,

$$\sum_{j \neq i} A_{\tau,ij} = 1 - A_{\tau,ii} \le (N-1)\exp(-\Delta_i/\tau). \tag{4}$$

Proofs are given in Appendix B. Increasing τ tightens the upper bound on within-row weight contrast. For a fixed positive similarity gap, decreasing τ tightens the upper bound on total cross-channel weight. At finite temperatures, the softmax graph remains dense.

The two bounds also give the limiting interaction patterns. The right-hand side of Eq. (4) approaches zero as $\tau \to 0^+$. As $\tau \to \infty$, the weight-ratio bound approaches one. Row stochasticity then gives

$$\lim_{\tau \to 0^+} \mathbf{A}_\tau = \mathbf{I}, \qquad \lim_{\tau \to \infty} \mathbf{A}_\tau = \mathbf{U} := \frac{1}{N}\mathbf{1}\mathbf{1}^\top, \tag{5}$$

where the low-temperature limit requires $\Delta_i > 0$ in every row. These limits describe the temperature scaling of fixed similarities; they do not assert a monotonic trajectory across models trained separately at different temperatures.

The adjacency depends jointly on temperature and the learned channel similarities. To understand how temperature scaling is expressed in the resulting interaction pattern, we examine self-propagation and the allocation of weights among other channels. Since each row sums to one, diagonal weights near one imply little interaction with other channels, giving CI-like behavior. When interactions are present, CP emphasizes selected relations while suppressing others, whereas CD allows broader interaction (Figure 2(c)). Such selectivity is reflected in uneven weights among the off-diagonal entries of the adjacency.

However, the full matrix can also be nonuniform because of large diagonal weights, even when the off-diagonal weights within each row are equal. We therefore account for average self-propagation before characterizing the remaining structure, using identity propagation $\mathbf{I}$ and uniform mixing $\mathbf{U} = \mathbf{1}\mathbf{1}^\top/N$ as joint structural references. Their combinations adjust the balance between self-propagation and uniform mixing. We seek the closest combination to the adjacency $\mathbf{A} = \mathbf{A}_\tau$ under the Frobenius norm. With a coefficient $\alpha \in \mathbb{R}$ and residual $\mathbf{R}(\alpha)$, this gives

$$\mathbf{A} = \alpha\mathbf{I} + (1-\alpha)\mathbf{U} + \mathbf{R}(\alpha). \tag{6}$$

Theorem 2 (Identity–uniform projection)**.** *For a row-stochastic $\mathbf{A} \in \mathbb{R}^{N\times N}$ with $N > 1$, the unique coefficient minimizing $\|\mathbf{R}(\alpha)\|_F^2$ satisfies*

$$\widehat{\alpha} = \frac{\mathrm{tr}(\mathbf{A}) - 1}{N-1}, \qquad \langle \mathbf{R}(\widehat{\alpha}), \mathbf{I} - \mathbf{U}\rangle_F = 0. \tag{7}$$

Proof. The projection condition is $\langle \mathbf{R}(\alpha), \mathbf{I}-\mathbf{U}\rangle_F = 0$. Since $\mathbf{R}(\alpha) = \mathbf{A} - \mathbf{U} - \alpha(\mathbf{I}-\mathbf{U})$, row stochasticity gives $\langle \mathbf{R}(\alpha), \mathbf{I}-\mathbf{U}\rangle_F = \mathrm{tr}(\mathbf{A}) - 1 - \alpha(N-1)$. The squared residual norm is strictly convex in α because $\|\mathbf{I}-\mathbf{U}\|_F^2 = N-1 > 0$, so the stated coefficient is its unique minimizer. □

We call $\widehat{\alpha}$ the *Identity Strength*. It measures average self-propagation relative to the two references, taking values one at $\mathbf{I}$ and zero at $\mathbf{U}$.

The *Interaction Heterogeneity* summarizes the structure remaining in $\mathbf{R} = \mathbf{R}(\widehat{\alpha})$ through its normalized energy:

$$\mathrm{IS}(\mathbf{A}) = \widehat{\alpha}, \qquad \mathrm{IH}(\mathbf{A}) = \mathcal{S}(\mathbf{A}) = \frac{\|\mathbf{R}\|_F^2}{N-1}. \tag{8}$$

Proposition 1 (Pythagorean decomposition)**.** *The diagnostics in Eq. (8) satisfy*

$$\frac{\|\mathbf{A}-\mathbf{U}\|_F^2}{N-1} = \mathrm{IS}(\mathbf{A})^2 + \mathrm{IH}(\mathbf{A}). \tag{9}$$

The proof is given in Appendix C. The two terms measure the normalized energies of $\mathbf{A} - \mathbf{U}$ along the identity–uniform direction and in its orthogonal complement, respectively. Comparable Identity Strength can coexist with different levels of Interaction Heterogeneity, exposing differences in relation weighting beyond average self-propagation. The low and high temperature limits in Eq. (5) correspond to $(\mathrm{IS}, \mathrm{IH}) = (1, 0)$ and $(0, 0)$, respectively. The structural diagnostics describe the base adjacency $\mathbf{A}$. These metrics describe learned interaction patterns across temperatures and compare fitted graphs across datasets in Section 4.4. They do not enter the training objective.

3.3 τ-Graph Temporal Attention

Using this graph, we propagate queries across channels (Figure 2(a)), allowing related channels to guide historical patch selection. Given the input $\mathbf{H}^{(\ell-1)} \in \mathbb{R}^{B\times N\times M\times D}$ to the ℓ-th block, the query, key, and value representations are obtained through linear projections:

$$\mathbf{Q}^{(\ell)} = \mathbf{H}^{(\ell-1)}\mathbf{W}_Q^{(\ell)}, \quad \mathbf{K}^{(\ell)} = \mathbf{H}^{(\ell-1)}\mathbf{W}_K^{(\ell)}, \quad \mathbf{V}^{(\ell)} = \mathbf{H}^{(\ell-1)}\mathbf{W}_V^{(\ell)}. \tag{10}$$

The projections act on the feature dimension; bias terms are omitted here for clarity. These projections are recomputed in every block, rather than taking keys and values from the initial embedding at all depths.

A simplified GCN uses the learned adjacency $\mathbf{A}_\tau^{(\ell)}$ to aggregate queries across channels at each patch position over k supports defined by order-i Chebyshev polynomials T_i (Defferrard et al., 2016):

$$\widetilde{\mathbf{Q}}_{b,:,m,:}^{(\ell)} = \sum_{i=0}^{k-1} \mathbf{G}_i^{(\ell)} \mathbf{Q}_{b,:,m,:}^{(\ell)}, \qquad \mathbf{G}_i^{(\ell)} = T_i\Big(\mathbf{A}_\tau^{(\ell)}\Big). \tag{11}$$

At patch position m in sample b, the colons select all channels and features, yielding an $N \times D$ query matrix. Following LightGCN (He et al., 2020), graph propagation uses no additional feature transformation or nonlinear activation. The underlying channel embeddings remain trainable.

Standard temporal attention then uses the enhanced queries with keys and values projected from each channel's current representation (Vaswani et al., 2017). For channel n and a head h of dimension D_h, attention is computed with softmax over key patches:

$$\mathbf{O}_{b,n}^{(\ell,h)} = \mathrm{Softmax}\left(\frac{\widetilde{\mathbf{Q}}_{b,n}^{(\ell,h)} (\mathbf{K}_{b,n}^{(\ell,h)})^\top}{\sqrt{D_h}}\right) \mathbf{V}_{b,n}^{(\ell,h)}. \tag{12}$$

The head outputs are concatenated and projected, followed by standard residual, normalization, and feed-forward operations.

3.4 Forecasting and Optimization

After L_e encoder blocks, the patch and feature dimensions are flattened and projected to the forecasting horizon:

$$\widehat{\widehat{\mathbf{y}}}_{b,n} = \mathbf{W}_o \,\mathrm{vec}\Big(\mathbf{H}_{b,n,:,:}^{(L_e)}\Big) + \mathbf{b}_o. \tag{13}$$

The predictions are transformed back to the original scale using the input normalization statistics.

We train τformer using the Huber loss ℓ (Huber, 1992) to measure discrepancies between predictions and ground truth:

$$\mathcal{L} = \frac{1}{BTN} \sum_{b=1}^{B} \sum_{t=1}^{T} \sum_{n=1}^{N} \ell\Big(\widehat{Y}_{b,t,n} - Y_{b,t,n}\Big). \tag{14}$$

Details of the forecasting pipeline and graph propagation, proofs of the theoretical results, and computational complexity analysis are provided in Appendices A–E.

4 Experiments

4.1 Experimental Setup

Datasets. We use nine datasets: the four ETT variants, Weather, Electricity, Traffic, Solar-Energy, and Exchange. Chronological splits are 12/4/4 months for ETT and 70%/10%/20% otherwise.

Baselines. We compare with ten baselines grouped by backbone: Transformer-based models, including SSformer (Liu et al., 2026), PCD (iTransformer backbone) (Lee et al., 2026), DUET (Qiu et al., 2025b), TimeXer (Wang et al., 2024b), iTransformer (Liu et al., 2024), and PatchTST (Nie et al., 2022); MLP-based DLinear (Zeng et al., 2023) and TimeMixer (Wang et al., 2024a); CNN-based TimesNet (Wu et al., 2023); and GNN-based MSGNet (Cai et al., 2024).

Implementations. Unless otherwise stated, we use an input length of 96 and prediction horizons of $\{96, 192, 336, 720\}$. The patch length and stride are 16 and 8, respectively. We implement τformer in PyTorch (Paszke et al., 2019) and train it on a single NVIDIA RTX A6000 GPU. Training uses Adam to optimize the Huber loss, with learning rate schedules set separately for each dataset. Batch sizes are selected from $\{16, 32, 128\}$ according to the number of variables in each dataset. We train for up to 10 epochs and retain the checkpoint with the lowest validation loss. The input/prediction lengths and the epoch limit follow the benchmark settings (Wang et al., 2026).

Table 1: Forecasting MSE and MAE on nine datasets. Avg. averages the four horizons. Best and second-best values at the reported precision are bold and underlined. 1st Count includes ties and Avg. rows. Lower is better.

Models		**τformer**		SSformer		PCD		DUET		MSGNet		TimeMixer		TimeXer		iTransformer		PatchTST		TimesNet		DLinear	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	.378	.395	**.373**	**.393**	.385	.404	.377	**.393**	.390	.411	.375	.400	.382	.403	.386	.405	.414	.419	.384	.402	.386	.400
	192	**.428**	.426	.432	.431	.438	.434	.429	.425	.442	.442	.429	**.421**	.429	.435	.441	.436	.460	.445	.436	.429	.437	.432
	336	**.467**	**.446**	.483	.454	.475	.454	.471	**.446**	.480	.468	.484	.458	.468	.448	.487	.458	.501	.466	.491	.469	.481	.459
	720	.478	.472	.519	.495	.477	.474	.496	.480	.494	.488	.498	.482	**.469**	**.461**	.503	.491	.500	.488	.521	.500	.519	.516
	Avg.	.438	**.435**	.452	.443	.444	.441	.443	.436	.452	.452	.447	.440	**.437**	.437	.454	.447	.469	.454	.458	.450	.456	.452
ETTh2	96	**.281**	**.330**	**.281**	.332	.295	.347	.296	.345	.328	.371	.289	.341	.286	.338	.297	.349	.302	.348	.340	.374	.333	.387
	192	**.360**	**.380**	.367	.385	.380	.397	.368	.389	.402	.414	.372	.392	.363	.389	.380	.400	.388	.400	.402	.414	.477	.476
	336	.401	**.414**	.394	.418	.427	.434	.411	.422	.435	.443	**.386**	**.414**	.414	.423	.428	.432	.426	.433	.452	.452	.594	.541
	720	.403	.427	**.395**	**.420**	.432	.445	.412	.434	.417	.441	.412	.434	.408	.432	.427	.445	.431	.446	.462	.468	.831	.657
	Avg.	.361	**.388**	**.359**	.389	.383	.406	.372	.397	.396	.417	.364	.395	.367	.396	.383	.407	.387	.407	.414	.427	.559	.515
ETTm1	96	**.313**	**.354**	.318	.355	.331	.369	.324	**.354**	.319	.366	.320	.357	.318	.356	.334	.368	.329	.367	.338	.375	.345	.372
	192	**.358**	**.379**	.360	**.379**	.372	.390	.369	**.379**	.376	.397	.361	.381	.362	.383	.377	.391	.367	.385	.374	.387	.380	.389
	336	**.387**	**.400**	.393	.402	.412	.414	.404	.402	.417	.422	.390	.404	.395	.407	.426	.420	.399	.410	.410	.411	.413	.413
	720	.459	.442	**.449**	.440	.479	.453	.463	**.437**	.481	.458	.454	.441	.452	.441	.491	.459	.454	.439	.478	.450	.474	.453
	Avg.	**.379**	.394	.380	.394	.398	.406	.390	**.393**	.398	.411	.381	.395	.382	.397	.407	.410	.387	.400	.400	.406	.403	.407
ETTm2	96	**.171**	**.250**	**.171**	.254	.184	.272	.174	.255	.177	.262	.175	.258	**.171**	.256	.180	.264	.175	.259	.187	.267	.193	.292
	192	.240	**.295**	**.235**	.302	.251	.311	.243	.302	.247	.307	.237	.299	.237	.299	.250	.309	.241	.302	.249	.309	.284	.362
	336	.303	**.335**	.307	.338	.312	.350	.304	.341	.312	.346	.298	.340	**.296**	.338	.311	.348	.305	.343	.321	.351	.369	.427
	720	.401	**.393**	.403	.395	.412	.408	.399	.397	.414	.403	**.391**	.396	.392	.394	.412	.407	.402	.400	.408	.403	.554	.522
	Avg.	.279	**.318**	.279	.322	.289	.335	.280	.324	.288	.330	.275	.323	**.274**	.322	.288	.332	.281	.326	.291	.333	.350	.401
Weather	96	**.156**	**.194**	**.156**	.204	.165	.209	.163	.202	.163	.212	.163	.209	.157	.205	.174	.214	.177	.218	.172	.220	.196	.255
	192	**.203**	**.239**	**.203**	.245	.213	.251	.218	.252	.212	.254	.208	.250	.204	.247	.221	.254	.225	.259	.219	.261	.237	.296
	336	.259	**.282**	**.250**	.284	.274	.296	.274	.294	.272	.299	.251	.287	.261	.290	.278	.296	.278	.297	.280	.306	.283	.335
	720	**.337**	**.332**	.338	.337	.350	.346	.349	.343	.350	.348	.339	.341	.340	.341	.358	.347	.354	.348	.365	.359	.345	.381
	Avg.	.239	**.262**	**.237**	.268	.250	.275	.251	.273	.249	.278	.240	.271	.241	.271	.258	.278	.259	.281	.259	.287	.265	.317
Electricity	96	**.139**	**.231**	.141	.238	.140	.235	.145	.233	.165	.274	.153	.247	.140	.242	.148	.240	.181	.270	.168	.272	.197	.282
	192	.159	**.248**	**.157**	.251	.158	.252	.163	**.248**	.184	.292	.166	.256	**.157**	.256	.162	.253	.188	.274	.184	.289	.196	.285
	336	**.171**	.264	.174	.268	.172	.267	.175	**.262**	.195	.302	.185	.277	.176	.275	.178	.269	.204	.293	.198	.300	.209	.301
	720	**.190**	**.280**	.209	.303	.202	.295	.204	.291	.231	.332	.225	.310	.211	.306	.225	.317	.246	.324	.220	.320	.245	.333
	Avg.	**.165**	**.256**	.170	.265	.168	.262	.172	.258	.194	.300	.182	.272	.171	.270	.178	.270	.205	.290	.192	.295	.212	.300
Traffic	96	**.381**	**.248**	.420	.288	.391	.266	.407	.252	.605	.344	.462	.285	.428	.271	.395	.268	.462	.295	.593	.321	.650	.396
	192	**.403**	**.258**	.434	.298	.409	.275	.431	.262	.613	.359	.473	.296	.448	.282	.417	.276	.466	.296	.617	.336	.598	.370
	336	**.424**	**.265**	.468	.301	.426	.282	.456	.269	.642	.376	.498	.296	.473	.289	.433	.283	.482	.304	.629	.336	.605	.373
	720	**.457**	**.285**	.487	.309	.460	.300	.509	.292	.702	.401	.506	.313	.516	.307	.467	.302	.514	.322	.640	.350	.645	.394
	Avg.	**.416**	**.264**	.452	.299	.422	.281	.451	.269	.641	.370	.484	.297	.466	.287	.428	.282	.481	.304	.620	.336	.625	.383
Solar-Energy	96	**.182**	**.205**	.197	.236	.197	.231	.200	.207	.208	.243	.189	.259	.215	.295	.203	.237	.234	.286	.250	.292	.290	.378
	192	**.222**	**.232**	**.222**	.268	.232	.260	.228	.233	.258	.281	**.222**	.283	.236	.301	.233	.261	.267	.310	.296	.318	.320	.398
	336	.239	.247	.245	.271	.241	.270	.262	**.244**	.293	.311	**.231**	.292	.252	.307	.248	.273	.290	.315	.319	.330	.353	.415
	720	.242	.254	.247	.287	.241	.273	.258	**.249**	.290	.315	**.223**	.285	.244	.305	.249	.275	.289	.317	.338	.337	.356	.413
	Avg.	.221	.235	.228	.266	.228	.258	.237	**.233**	.262	.288	**.216**	.280	.237	.302	.233	.262	.270	.307	.301	.319	.330	.401
Exchange	96	**.081**	**.198**	.085	.200	.085	.205	.082	.199	.102	.230	.090	.235	.087	.208	.086	.206	.088	.205	.107	.234	.088	.218
	192	**.171**	**.293**	**.171**	.296	.180	.302	.174	.295	.195	.317	.187	.343	.181	.302	.177	.299	.176	.299	.226	.344	.176	.315
	336	.318	.408	.348	.429	.337	.421	.326	.413	.359	.436	.353	.473	.338	.420	.331	.417	**.301**	**.397**	.367	.448	.313	.427
	720	**.816**	**.679**	.843	.715	.850	.696	.843	.692	.940	.738	.934	.761	.865	.702	.847	.691	.901	.714	.964	.746	.839	.695
	Avg.	**.347**	**.395**	.362	.410	.363	.406	.356	.400	.399	.430	.391	.453	.368	.408	.360	.403	.367	.404	.416	.443	.354	.414
1st Count		27	34	14	3	0	0	0	11	0	0	6	2	6	1	0	0	1	1	0	0	0	0

4.2 Main Results

Across the 45 rows in Table 1, including dataset averages, τformer achieves the best MSE in 27 and the best MAE in 34, counting ties at the reported precision. This supports adapting channel interaction through temperature-controlled propagation within a common graph construction. On Traffic, it ranks first in both metrics at all four horizons, with average MSE/MAE of 0.416/0.264, and outperforms every baseline by at least 6.0% in one or both of MSE and MAE. These gains are consistent with graph propagation enriching queries with cross-channel information (Section 3.3).

4.3 Ablation Study

We study these components because temperature and the adaptive graph form the core channel interaction mechanism, while temporal embeddings provide positional information for temporal attention. *w/o Temporal Emb.* removes the patch-position embedding module, including its dropout (Appendix A.2). *w/o Temperature* fixes $\tau = 1$, removing temperature control of channel interactions (Section 3.2). *w/o Adaptive Graph* sets $\mathbf{A} = \mathbf{I}$, eliminating cross-channel aggregation (Section 3.2).

Table 2 shows that removing temporal embeddings worsens both metrics except on Weather and Traffic. Fixing $\tau = 1$ worsens both metrics on Weather and Electricity but ties on Traffic. Graph removal causes the largest degradation on Weather, Electricity, Solar-Energy, and Traffic, supporting cross-channel aggregation.

Table 2: Ablation results, with MSE and MAE averaged over the four prediction horizons. *w/o* denotes removal of the corresponding component. Best values at the reported precision are bold.

Models	ETTh1		ETTh2		ETTm1		ETTm2		Exchange		Weather		Electricity		Solar-Energy		Traffic	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
τformer	**.438**	**.435**	**.361**	**.388**	**.379**	**.394**	**.279**	**.318**	**.347**	**.395**	**.239**	**.262**	**.165**	**.256**	**.221**	**.235**	**.416**	**.264**
w/o Temporal Emb.	.447	.441	.368	.392	.396	.404	.282	.321	.349	.398	**.239**	**.262**	.168	.260	.224	.236	**.416**	**.264**
w/o Temperature	.445	.436	.362	**.388**	.397	.403	.280	.320	.355	.402	.246	.266	.170	.261	.222	.236	**.416**	**.264**
w/o Adaptive Graph	**.438**	**.435**	.365	.390	.386	.395	.281	.320	.355	.401	.255	.270	.190	.271	.246	.249	.452	.274

4.4 Adaptive Relation Learning Across Datasets

We examine whether one graph construction adapts to different channel interaction requirements. Across the combinations of datasets and horizons in Figure 3, IS and IH, computed after averaging adjacency matrices across layers, place CI at the lower right, CP in the upper middle, and CD near the origin. Each dataset's pattern is largely stable across horizons under its chosen configuration.

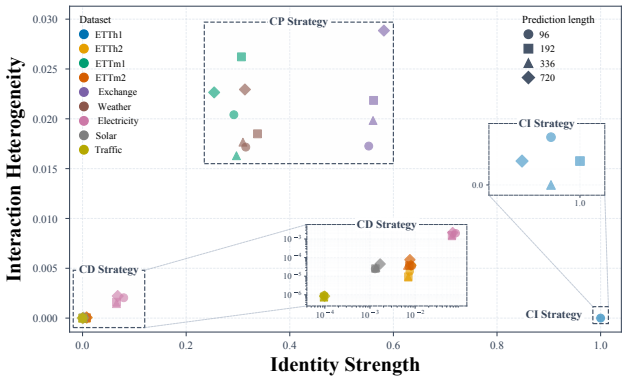

Figure 3: Adaptation of learned channel relations across datasets. A shared graph construction accommodates different interaction requirements through CI, CP, and CD. Each dataset's interaction pattern remains largely stable across prediction horizons under its chosen configuration.

These patterns support the unified graph view in Section 3.2, where temperature connects self-propagation and diffuse interactions. Under Theorem 2, IS measures average self-propagation relative to identity and uniform mixing; IH captures residual structure. Similar IS can accompany different IH, so IS alone cannot distinguish the three strategies. Channel embeddings learn relational preferences; temperature controls their concentration.

4.5 Temperature Control of Channel Strategies

We examine how temperature changes channel strategies and forecasting accuracy. Figure 4 shows results from separate training at each temperature. On ETTh1, low temperatures produce diagonal-dominated matrices and lower MSE, consistent with CI; MAE is nearly flat initially. Weather favors CP around $\tau = 0.1$, with unequal off-diagonal weights and lower MSE and MAE than at either temperature extreme. On ETTh2, both errors initially rise and then fall toward a plateau at high temperatures, where nearly uniform weights indicate CD.

The progression from CI through CP toward CD is consistent with the limits of identity propagation and uniform mixing for fixed similarities in Section 3.2; the low-temperature limit requires a unique

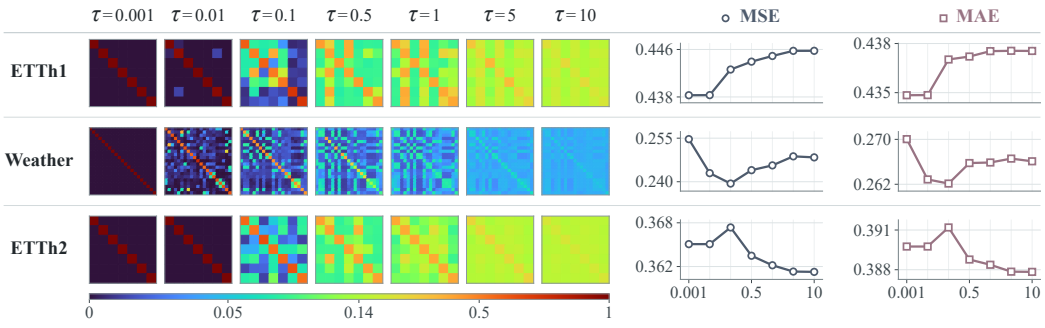

Figure 4: Adapting channel strategies to different datasets. Lower forecasting errors correspond to different strategies on ETTh1, Weather, and ETTh2. Temperature controls learned channel weight concentration, allowing the same graph construction to accommodate CI, CP, and CD.

diagonal maximum per row. No single strategy is preferable across these datasets. Temperature accommodates their different requirements by adjusting channel propagation and hence attention.

4.6 Sharper Relational Discrimination on Weather

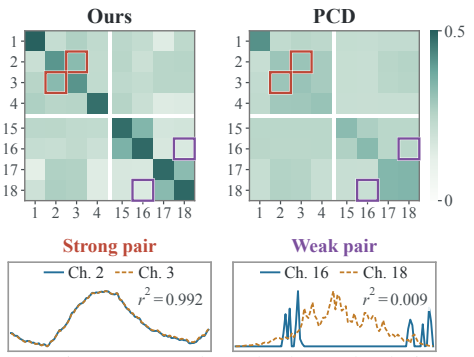

Figure 5: Strong and weak channel relations on Weather. Our adjacency distinguishes the highlighted pairs more sharply than PCD's attention. Raw directed weights share a nonlinear scale.

To examine how well τformer learns channel relations, we compare it with PCD (Lee et al., 2026), which also uses explicit channel relations to influence attention. On Weather, Figure 5 compares our average learned adjacency with PCD's average channel attention. These matrices represent graph propagation preferences and channel aggregation weights, respectively. At the selected directed entries, our adjacency assigns 49.8% higher weight to the strongly correlated pair and 82.5% lower weight to the weakly correlated pair than PCD's channel attention.

The sharper contrast is consistent with temperature controlling how strongly the adjacency reflects learned similarity preferences (Eq. (2)). PCD instead multiplies input-dependent attention scores elementwise by a learned mask before the softmax to yield channel weights. The effect of mask adjustment on the resulting interaction scope has limited theoretical characterization. For fixed similarities, Theorem 1 bounds the within-row weight ratios. A unique diagonal maximum in each row also yields bounds on total cross-channel weight per row, supporting direct concentration control.

4.7 Robust Forecasting with Learned Channel Relations

To test whether learned relations remain useful under noisy observations, we compare forecasting errors on ETTh2 (Figure 6). We add Gaussian noise sampled from $\mathcal{N}(0, \sigma^2)$ only to standardized test inputs. τformer has the lowest average MAE at every tested noise level; from $\sigma = 0$ to 0.4, its MAE increases by 6.5%, versus at least 8.2% for the competing methods. Thus, τformer combines lower error under corruption with smaller relative degradation.

These results support using channel context for temporal pattern selection (Section 3.3). Graph propagation incorporates channel context into the queries, which attend to each channel's own keys and values (Eqs. (11) and (12)). The observed robustness suggests that relations learned from clean data can still guide the selection of historical patterns from noisy observations.

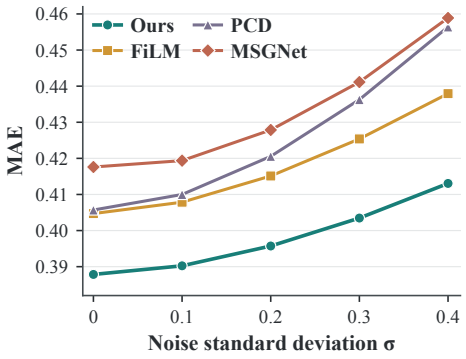

Figure 6: Forecasting with learned channel relations is accurate under input corruption. On ETTh2, τformer has the lowest MAE at all tested noise levels with broad channel interaction.

5 Conclusion

We introduced τformer, which accommodates CI, CP, and CD through temperature scaling of a learned channel graph, making interaction scope adjustable within the same forecasting architecture. We derive temperature-dependent bounds on weight contrast and total cross-channel weight, and introduce Identity Strength and Interaction Heterogeneity to characterize average self-propagation and residual interaction structure. Experimental results show competitive forecasting performance, supporting the value of adjusting channel interaction scope.

AI USE STATEMENT

We used generative AI tools, including ChatGPT and Codex, to aid or polish writing, to draft sections of the paper, and to assist with proving mathematical claims.

For the tasks requiring disclosure under the ICLR 2027 policy, AI was not used for retrieval and discovery, research ideation or execution, or generating synthetic datasets.

The authors checked all AI-assisted outputs. In particular, AI-assisted mathematical arguments and proofs were independently verified by the authors, and AI-assisted writing was reviewed and revised before inclusion in the paper. The authors accept responsibility for all content and results presented in this paper.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Maximilian Beichter, Nils Friederich, Janik Pinter, Dorina Werling, Kaleb Phipps, Sebastian Beichter, Oliver Neumann, Ralf Mikut, Veit Hagenmeyer, and Benedikt Heidrich. Decision-focused fine-tuning of time series foundation models for dispatchable feeder optimization. *Energy and AI*, pp. 100533, 2025.

Abdelhakim Benechehab, Vasilii Feofanov, Giuseppe Paolo, Albert Thomas, Maurizio Filippone, and Balázs Kégl. AdaPTS: Adapting univariate foundation models to probabilistic multivariate time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=yeICCRy3lE.

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 11141–11149, 2024.

Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassiulas, Jure Leskovec, and Rex Ying. From similarity to superiority: Channel clustering for time series forecasting. *Advances in Neural Information Processing Systems*, 37:130635–130663, 2024a.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 10527–10545, 2024b.

Hanyin Cheng, Ruitong Zhang, Yuning Lu, Peng Chen, Meng Wang, Yang Shu, Bin Yang, and Chenjuan Guo. Star: Boosting time series foundation models for anomaly detection through state-aware adapter. *arXiv preprint arXiv:2510.16014*, 2025.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H. Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (ttms): fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, 2024.

Matthew Faw, Rajat Sen, Yichen Zhou, and Abhimanyu Das. In-context fine-tuning for time-series foundation models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=uxzgGLWPj2.

Vasilii Feofanov, Songkang Wen, Marius Alonso, Romain Ilbert, Hongbo Guo, Malik Tiomoko, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight calibrated foundation model for user-friendly time series classification. *arXiv preprint arXiv:2502.15637*, 2025.

CW Fu and TT Nguyen. Models for long-term energy forecasting. In *2003 IEEE Power Engineering Society General Meeting (IEEE Cat. No. 03CH37491)*, volume 1, pp. 235–239. IEEE, 2003.

Haotian Gao, Zheng Dong, Jiawei Yong, Shintaro Fukushima, Kenjiro Taura, and Renhe Jiang. How different from the past? spatio-temporal time series forecasting with self-supervised deviation learning. *Advances in Neural Information Processing Systems*, 38:146647–146675, 2026.

Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yiluan Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *Forty-first International Conference on Machine Learning (ICML)*, 2024.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *Advances in Neural Information Processing Systems*, 37: 64145–64175, 2024.

Lu Han, Yu Liu, Qiwen Deng, Jian Jiang, Yinbo Sun, Zhe Yu, Binfeng Wang, Xingyu Lu, Lintao Ma, Han-Jia Ye, et al. Unica: Adapting time series foundation model to general covariate-aware forecasting. *arXiv preprint arXiv:2506.22039*, 2025.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, 2020.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=490VcNtjh7.

Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Seunghan Lee, Taeyoung Park, and Kibok Lee. Dataset-driven channel masks in transformers for multivariate time series. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2401–2405. IEEE, 2026.

Dongyuan Li, Shun Zheng, Chang Xu, Jiang Bian, and Renhe Jiang. Routing channel-patch dependencies in time series forecasting with graph spectral decomposition. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=uIPAuyno4Z.

Ying Liu, Bo Liu, Sheng Huang, Gang Luo, Wenbo Hu, Meng Wang, and Richang Hong. Sparse-scale transformer with bidirectional awareness for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23899–23907, 2026.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In *Forty-second International Conference on Machine Learning*, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Guo Qin, Zhi Chen, Yong Liu, Zhiyuan Shi, Haixuan Liu, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Cora: Covariate-aware adaptation of time series foundation models. *arXiv preprint arXiv:2510.12681*, 2025.

Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Junkai Lu, Jilin Hu, Chenjuan Guo, Christian S Jensen, and Bin Yang. A comprehensive survey of deep learning for multivariate time series forecasting: A channel strategy perspective. *arXiv preprint arXiv:2502.10721*, 2025a.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1185–1196, 2025b.

Licheng Qu, Wei Li, Wenjing Li, Dongfang Ma, and Yinhai Wang. Daily long-term traffic flow forecasting based on a deep neural network. *Expert Systems with applications*, 121:304–312, 2019.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021.

Xiaohou Shi, Ke Li, Aobo Liang, and Yan Sun. Time tracker: Mixture-of-experts-enhanced foundation time series forecasting model with decoupled training pipelines. *arXiv preprint arXiv:2505.15151*, 2025.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts, 2024. URL `https://arxiv.org/abs/2409.16040`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 38626–38652, 2024a.

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*, volume 2025, pp. 16980–17016, 2025a.

Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, et al. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. In *Forty-second International Conference on Machine Learning*.

Yihang Wang, Yuying Qiu, Peng Chen, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Towards a general time series forecasting model with unified representation and adaptive transfer. In *Forty-second International Conference on Machine Learning*, 2025b.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024b.

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*, 2026.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Congxi Xiao, Jingbo Zhou, Yixiong Xiao, Xinjiang Lu, Le Zhang, and Hui Xiong. Timefound: A foundation model for time series forecasting. *arXiv preprint arXiv:2503.04118*, 2025.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in neural information processing systems*, 36:69638–69660, 2023.

Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S. Yu. How much can transfer? BRIDGE: Bounded multi-domain graph foundation model with generalization guarantees. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=bjDKZ3Roax.

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.

Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In *International Conference on Learning Representations*, volume 2024, pp. 41433–41451, 2024.

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.

Tian Zhou, Ziqing Ma, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. FiLM: Frequency improved Legendre memory model for long-term time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

A Detailed Forecasting Pipeline

A.1 Instance Normalization

Following the series stationarization approach (Liu et al., 2022), we normalize each input instance along the temporal dimension and restore its statistics after forecasting. For an input batch $\mathbf{X} \in \mathbb{R}^{B\times L\times N}$, the temporal mean and variance of channel n in sample b are

$$\mu_{b,n} = \frac{1}{L}\sum_{l=1}^{L} X_{b,l,n}, \qquad \sigma_{b,n}^2 = \frac{1}{L}\sum_{l=1}^{L}\left(X_{b,l,n} - \mu_{b,n}\right)^2. \tag{15}$$

The normalized input is

$$\widetilde{X}_{b,l,n} = \frac{X_{b,l,n} - \mu_{b,n}}{\sqrt{\sigma_{b,n}^2 + \epsilon}}. \tag{16}$$

The normalization statistics are detached from gradient computation and retained for de-normalization.

A.2 Patching and Embedding

Replication padding of length S is appended to the temporal dimension. Given patch length P and stride S, the number of patches is

$$M = \left\lfloor \frac{L + S - P}{S} \right\rfloor + 1. \tag{17}$$

The resulting patch tensor has shape $\mathbb{R}^{B\times N\times M\times P}$. Each patch is projected into $\mathbb{R}^D$ and combined with a learnable patch-position embedding:

$$\mathbf{H}_{b,n,m}^{(0)} = \mathbf{W}_p \mathbf{p}_{b,n,m} + \mathbf{b}_p + \mathbf{e}_m^{\mathrm{pos}}. \tag{18}$$

Here, $\mathbf{p}_{b,n,m} \in \mathbb{R}^P$ is a normalized input patch, $\mathbf{W}_p \in \mathbb{R}^{D\times P}$ is shared across channels, and $\mathbf{e}_m^{\mathrm{pos}} \in \mathbb{R}^D$ encodes its temporal position. Dropout is applied before the first encoder block.

A.3 Forecasting and De-Normalization

After the final graph-enhanced attention block, the representation $\mathbf{H}^{(L_e)} \in \mathbb{R}^{B\times N\times M\times D}$ is flattened along the patch and feature dimensions. The shared forecasting head produces a normalized prediction for each channel:

$$\widehat{\widetilde{\mathbf{y}}}_{b,n} = \mathbf{W}_o \,\mathrm{vec}\left(\mathbf{H}_{b,n,:,:}^{(L_e)}\right) + \mathbf{b}_o. \tag{19}$$

The original scale is restored by

$$\widehat{Y}_{b,t,n} = \sqrt{\sigma_{b,n}^2 + \epsilon}\,\widehat{\widetilde{Y}}_{b,t,n} + \mu_{b,n}. \tag{20}$$

A.4 Huber Training Objective

Let $e_{b,t,n} = \widehat{Y}_{b,t,n} - Y_{b,t,n}$ denote the forecasting error. The main text writes the Huber penalty as $\ell(\cdot)$; here we make its fixed threshold $\delta > 0$ explicit, so $\ell(e) = \ell_\delta(e)$. The penalty (Huber, 1992) is defined as

$$\ell_\delta(e) = \begin{cases} \dfrac{1}{2}e^2, & |e| \le \delta, \\ \delta\left(|e| - \dfrac{1}{2}\delta\right), & |e| > \delta. \end{cases} \tag{21}$$

The quadratic region encourages accurate fitting for small forecasting errors, while the linear region reduces sensitivity to occasional large deviations. The loss averages over samples, future timestamps, and channels:

$$\mathcal{L} = \frac{1}{BTN}\sum_{b=1}^{B}\sum_{t=1}^{T}\sum_{n=1}^{N}\ell_\delta(e_{b,t,n}). \tag{22}$$

B PROOFS OF TEMPERATURE-CONTROLLED GRAPH PROPERTIES

We analyze one layer with fixed nonzero embeddings and finite $N > 1$. The layer index is omitted, and $A_{\tau,ij}$ denotes entry (i, j) of $\mathbf{A}_\tau$, equivalently written as $A_{ij}^{(\tau)}$. These limits describe temperature scaling of fixed similarities; they do not assert a monotone trajectory across models trained separately at different temperatures.

B.1 ROW-WISE WEIGHT-RATIO BOUND

For any two entries in row i, the common softmax denominator cancels:

$$\frac{A_{\tau,ij}}{A_{\tau,ir}} = \exp\left(\frac{C_{ij} - C_{ir}}{\tau}\right). \tag{23}$$

Since the normalized cosine similarities are passed through ReLU, $C_{ij}, C_{ir} \in [0, 1]$. Therefore,

$$C_{ij} - C_{ir} \leq 1, \tag{24}$$

and hence

$$\frac{A_{\tau,ij}}{A_{\tau,ir}} \leq \exp\left(\frac{1}{\tau}\right). \tag{25}$$

Selecting the maximum and minimum entries within each row gives

$$\frac{\max_j A_{\tau,ij}}{\min_j A_{\tau,ij}} \leq \exp(1/\tau). \tag{26}$$

B.2 LOW-TEMPERATURE LIMIT

For nonzero channel embeddings, normalization gives $C_{ii} = 1$. Define the similarity gap

$$\Delta_{ij} = 1 - C_{ij}, \qquad j \neq i. \tag{27}$$

The diagonal and off-diagonal weights can be written as

$$A_{\tau,ii} = \frac{1}{1 + \sum_{j\neq i} \exp(-\Delta_{ij}/\tau)}, \tag{28}$$

and

$$A_{\tau,ij} = \frac{\exp(-\Delta_{ij}/\tau)}{1 + \sum_{r\neq i} \exp(-\Delta_{ir}/\tau)}. \tag{29}$$

Assume that the diagonal entry is the unique maximum in row i, and let

$$\Delta_i = 1 - \max_{j\neq i} C_{ij} > 0. \tag{30}$$

Then

$$\begin{aligned} 1 - A_{\tau,ii} &\leq \sum_{j\neq i} \exp(-\Delta_{ij}/\tau) \\ &\leq (N-1)\exp(-\Delta_i/\tau). \end{aligned} \tag{31}$$

The right-hand side approaches zero as $\tau \to 0$. Therefore, $A_{\tau,ii} \to 1$ and $A_{\tau,ij} \to 0$ for every $j \neq i$. Applying the result row by row gives

$$\lim_{\tau\to 0} \mathbf{A}_\tau = \mathbf{I}. \tag{32}$$

If multiple entries attain the same maximum similarity, the limiting mass is shared among those entries. Thus, the unique-maximum assumption is required for convergence to the identity matrix.

Residual channel interactions. Equation (29) shows that

$$A_{\tau,ij} = \mathcal{O}\left(\exp\left(-\frac{1 - C_{ij}}{\tau}\right)\right). \tag{33}$$

Only channels whose similarities are within $\mathcal{O}(\tau)$ of the self-similarity retain non-negligible interactions.

For nearby normalized embeddings with positive cosine similarity, the ReLU is inactive, so $C_{ij} = \cos\theta_{ij}$ and

$$1 - \cos\theta_{ij} \approx \frac{\theta_{ij}^2}{2}. \tag{34}$$

Consequently,

$$\exp\left(-\frac{1 - C_{ij}}{\tau}\right) \approx \exp\left(-\frac{\theta_{ij}^2}{2\tau}\right), \tag{35}$$

which resembles a local Gaussian kernel on the normalized embedding space with bandwidth proportional to $\sqrt{\tau}$.

B.3 High-Temperature Limit

For large τ, the exponential function admits the expansion

$$\exp(C_{ij}/\tau) = 1 + \frac{C_{ij}}{\tau} + \mathcal{O}(\tau^{-2}). \tag{36}$$

Let

$$\overline{C}_i = \frac{1}{N}\sum_{j=1}^{N} C_{ij} \tag{37}$$

be the row-wise mean similarity. The denominator becomes

$$\sum_{r=1}^{N} \exp(C_{ir}/\tau) = N + \frac{N\overline{C}_i}{\tau} + \mathcal{O}(\tau^{-2}). \tag{38}$$

Using the first-order reciprocal expansion gives

$$A_{\tau,ij} = \frac{1}{N} + \frac{C_{ij} - \overline{C}_i}{N\tau} + \mathcal{O}(\tau^{-2}). \tag{39}$$

Therefore,

$$\lim_{\tau\to\infty} \mathbf{A}_\tau = \frac{1}{N}\mathbf{1}\mathbf{1}^\top. \tag{40}$$

Equation (39) further shows that deviations from uniform mixing decay at rate $\mathcal{O}(1/\tau)$.

C Proof of the Identity–Uniform Projection

Throughout this section, $\mathbf{A}$ is row-stochastic and $N > 1$. For $\mathbf{R}(\alpha) = \mathbf{A} - \alpha\mathbf{I} - (1-\alpha)\mathbf{U}$, orthogonality to $\mathbf{I} - \mathbf{U}$ is the normal equation of the following one-dimensional projection. We establish the unique minimizer and its orthogonal residual in Theorem 2.

Let

$$\mathbf{U} = \frac{1}{N}\mathbf{1}\mathbf{1}^\top \tag{41}$$

and consider the affine family

$$\mathbf{B}(\alpha) = \alpha\mathbf{I} + (1-\alpha)\mathbf{U} = \mathbf{U} + \alpha(\mathbf{I} - \mathbf{U}). \tag{42}$$

For a learned row-stochastic adjacency matrix $\mathbf{A}$, we define

$$\widehat{\alpha} = \arg\min_{\alpha\in\mathbb{R}} \|\mathbf{A} - \mathbf{B}(\alpha)\|_F^2. \tag{43}$$

Since Eq. (43) is a one-dimensional least-squares projection,

$$\widehat{\alpha} = \frac{\langle \mathbf{A} - \mathbf{U}, \mathbf{I} - \mathbf{U} \rangle_F}{\|\mathbf{I} - \mathbf{U}\|_F^2}. \tag{44}$$

Because $\mathbf{A}$ is row-stochastic,

$$\begin{aligned} \langle \mathbf{A}, \mathbf{U} \rangle_F &= \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} = 1, \\ \langle \mathbf{I}, \mathbf{U} \rangle_F &= 1, \\ \|\mathbf{U}\|_F^2 &= 1. \end{aligned} \tag{45}$$

It follows that

$$\begin{aligned} \langle \mathbf{A} - \mathbf{U}, \mathbf{I} - \mathbf{U} \rangle_F &= \mathrm{tr}(\mathbf{A}) - 1, \\ \|\mathbf{I} - \mathbf{U}\|_F^2 &= N - 1. \end{aligned} \tag{46}$$

Because $\|\mathbf{I} - \mathbf{U}\|_F^2 = N - 1 > 0$, the squared residual norm is strictly convex in α, so this minimizer is unique. Substitution into Eq. (44) yields

$$\widehat{\alpha} = \frac{\mathrm{tr}(\mathbf{A}) - 1}{N - 1}. \tag{47}$$

Define the residual matrix as

$$\mathbf{R} = \mathbf{A} - [\widehat{\alpha}\mathbf{I} + (1 - \widehat{\alpha})\mathbf{U}]. \tag{48}$$

By the normal equation of the least-squares projection,

$$\langle \mathbf{R}, \mathbf{I} - \mathbf{U} \rangle_F = 0. \tag{49}$$

Therefore,

$$\mathbf{A} = \widehat{\alpha}\mathbf{I} + (1 - \widehat{\alpha})\mathbf{U} + \mathbf{R} \tag{50}$$

is an orthogonal decomposition with respect to the identity–uniform direction.

The coefficient satisfies

$$\mathbf{A} = \mathbf{I} \Longrightarrow \widehat{\alpha} = 1, \qquad \mathbf{A} = \mathbf{U} \Longrightarrow \widehat{\alpha} = 0. \tag{51}$$

Thus, unlike the raw average diagonal value $\mathrm{tr}(\mathbf{A})/N$, $\widehat{\alpha}$ maps the uniform graph to zero and the identity graph to one for any number of channels.

The normalized residual energy is

$$\mathcal{S}(\mathbf{A}) = \frac{\|\mathbf{R}\|_F^2}{N - 1}. \tag{52}$$

A small value indicates that the graph is well approximated by a mixture of identity propagation and uniform mixing. Larger values indicate more remaining structure beyond average self-propagation.

C.1 Identity Strength and Interaction Heterogeneity

Let $\overline{d} = \mathrm{tr}(\mathbf{A})/N$. The *Identity Strength* is equivalently

$$\widehat{\alpha} = \frac{N\overline{d} - 1}{N - 1}. \tag{53}$$

For our graph construction, $C_{ii} = 1 \geq C_{ij}$ gives $A_{\tau,ii} \geq 1/N$. Hence $1 \leq \mathrm{tr}(\mathbf{A}) \leq N$ and $0 \leq \widehat{\alpha} \leq 1$. This interval relies on the diagonal preference of the cosine graph, not on row stochasticity alone.

Row stochasticity and the projected coefficient also imply

$$\mathbf{R}\mathbf{1} = \mathbf{0}, \qquad \mathrm{tr}(\mathbf{R}) = 0, \qquad \langle \mathbf{R}, \mathbf{I} - \mathbf{U} \rangle_F = 0. \tag{54}$$

The *Interaction Heterogeneity* is the normalized projection error:

$$\mathcal{S}(\mathbf{A}) = \frac{\|\mathbf{R}\|_F^2}{N - 1} = \frac{1}{N - 1} \min_{\alpha \in \mathbb{R}} \|\mathbf{A} - [\alpha\mathbf{I} + (1 - \alpha)\mathbf{U}]\|_F^2. \tag{55}$$

Since $\mathbf{A} - \mathbf{U} = \widehat{\alpha}(\mathbf{I} - \mathbf{U}) + \mathbf{R}$, the cross term vanishes. Using $\|\mathbf{I} - \mathbf{U}\|_F^2 = N - 1$ yields

$$\frac{\|\mathbf{A} - \mathbf{U}\|_F^2}{N-1} = \widehat{\alpha}^2 + \mathcal{S}(\mathbf{A}). \tag{56}$$

The two diagnostics must be interpreted together: $\widehat{\alpha}$ locates the projection along the identity–uniform axis, while $\mathcal{S}(\mathbf{A})$ measures departures from that axis. They do not uniquely determine the full adjacency matrix.

By continuity, the temperature limits proved above give

$$\tau \to 0 : \ (\widehat{\alpha}, \mathcal{S}) \to (1, 0), \qquad \tau \to \infty : \ (\widehat{\alpha}, \mathcal{S}) \to (0, 0), \tag{57}$$

where the first limit requires a unique diagonal maximum in every row. Nonzero heterogeneity identifies selective structure beyond a direct identity–uniform interpolation; it is not a hard clustering rule or a guarantee that every intermediate temperature produces CP-like behavior.

D Detailed Parameter-Free Graph Propagation

The current block input $\mathbf{H}^{(\ell-1)} \in \mathbb{R}^{B \times N \times M \times D}$ produces queries, keys, and values through affine feature projections:

$$\begin{aligned} \mathbf{Q}^{(\ell)} &= \mathbf{H}^{(\ell-1)}\mathbf{W}_Q^{(\ell)} + \mathbf{b}_Q^{(\ell)}, \\ \mathbf{K}^{(\ell)} &= \mathbf{H}^{(\ell-1)}\mathbf{W}_K^{(\ell)} + \mathbf{b}_K^{(\ell)}, \\ \mathbf{V}^{(\ell)} &= \mathbf{H}^{(\ell-1)}\mathbf{W}_V^{(\ell)} + \mathbf{b}_V^{(\ell)}. \end{aligned} \tag{58}$$

The main-text projections suppress the bias terms, which are made explicit here. Projection matrices act on the last dimension; biases are broadcast over samples, channels, and patches. These projections are recomputed in every block, rather than taking keys and values from the initial embedding at all depths.

For the ℓ-th graph-enhanced attention block, the adaptive adjacency matrix is denoted by $\mathbf{A}_\tau^{(\ell)}$. We use the Chebyshev polynomial basis (Defferrard et al., 2016), evaluated directly on the learned adjacency rather than the rescaled graph Laplacian used in the original spectral formulation. Consistent with Eq. (11), define $\mathbf{G}_i^{(\ell)} = T_i(\mathbf{A}_\tau^{(\ell)})$ for $i = 0, \ldots, k-1$. The graph supports are initialized as

$$\mathbf{G}_0^{(\ell)} = \mathbf{I}, \qquad \mathbf{G}_1^{(\ell)} = \mathbf{A}_\tau^{(\ell)}, \tag{59}$$

and the Chebyshev recurrence

$$\mathbf{G}_i^{(\ell)} = 2\mathbf{A}_\tau^{(\ell)}\mathbf{G}_{i-1}^{(\ell)} - \mathbf{G}_{i-2}^{(\ell)}, \qquad i = 2, \ldots, k-1. \tag{60}$$

Here $k \geq 2$, as in the evaluated configuration. For sample b and patch position m, the projected query tensor $\mathbf{Q}_{b,:,m,:}^{(\ell)} \in \mathbb{R}^{N \times D}$ is propagated as

$$\widetilde{\mathbf{Q}}_{b,:,m,:}^{(\ell)} = \sum_{i=0}^{k-1} \mathbf{G}_i^{(\ell)} \mathbf{Q}_{b,:,m,:}^{(\ell)}. \tag{61}$$

A conventional graph convolution is commonly written as

$$\mathbf{X}' = \mathbf{A}\mathbf{X}\mathbf{W}, \tag{62}$$

where $\mathbf{W}$ is an additional trainable feature transformation. Following the simplified propagation principle of LightGCN (He et al., 2020), τformer fixes the corresponding transformation to the identity:

$$\mathbf{W}_i = \mathbf{I}_D, \qquad i = 0, \ldots, k-1. \tag{63}$$

Therefore,

$$T_i\left(\mathbf{A}_\tau\right)\mathbf{Q}\mathbf{W}_i = T_i\left(\mathbf{A}_\tau\right)\mathbf{Q}, \tag{64}$$

which gives Eq. (61). The graph operation introduces no additional trainable feature transformation and only performs multi-order channel aggregation. The trainable query projection $\mathbf{W}_Q$ is applied

before graph propagation and is part of the standard attention projections rather than the graph module. The channel embeddings that define the adjacency remain trainable.

After channel propagation, the graph-enhanced query is reshaped into multiple attention heads. Temporal attention is applied independently to each channel over the M patch positions:

$$\mathbf{O}_{b,n}^{(\ell,h)} = \mathrm{Softmax}\left(\frac{\widetilde{\mathbf{Q}}_{b,n}^{(\ell,h)}\left(\mathbf{K}_{b,n}^{(\ell,h)}\right)^{\top}}{\sqrt{D_h}}\right)\mathbf{V}_{b,n}^{(\ell,h)}. \tag{65}$$

Softmax is taken over key patches. The head outputs are concatenated and linearly projected to $\mathbf{O}^{(\ell)}$, then passed through residual, normalization, and feed-forward operations:

$$\begin{aligned}\mathbf{Z}^{(\ell)} &= \mathrm{Norm}\left(\mathbf{H}^{(\ell-1)} + \mathrm{Dropout}\left(\mathbf{O}^{(\ell)}\right)\right),\\ \mathbf{H}^{(\ell)} &= \mathrm{Norm}\left(\mathbf{Z}^{(\ell)} + \mathrm{Dropout}\left(\mathrm{FFN}\left(\mathbf{Z}^{(\ell)}\right)\right)\right).\end{aligned} \tag{66}$$

Adjacency diagnostics and the propagation operator. The structural diagnostics describe the base adjacency $\mathbf{A}$. With the layer index omitted, the effective GCN operator is $p(\mathbf{A}) = \sum_{i=0}^{k-1}\mathbf{G}_i = \sum_{i=0}^{k-1} T_i(\mathbf{A})$, which need not be row-stochastic. In the evaluated three-support configuration,

$$p(\mathbf{A}) = \mathbf{I} + \mathbf{A} + (2\mathbf{A}^2 - \mathbf{I}) = \mathbf{A} + 2\mathbf{A}^2. \tag{67}$$

Consequently, $p(\mathbf{I}) = 3\mathbf{I}$ and $p(\mathbf{U}) = 3\mathbf{U}$. The adjacency endpoints therefore retain their self-propagation and global mixing interpretations in the GCN, up to query scaling. This scaling can affect attention concentration, so these are CI-like and CD-like interaction regimes rather than identities with specific baseline models.

E Computational Complexity

Constructing the embedding-based similarity matrix requires

$$\mathcal{O}(N^2 D_e) \tag{68}$$

operations per layer. Explicit construction of the dense supports $\mathbf{G}_0^{(\ell)}, \ldots, \mathbf{G}_{k-1}^{(\ell)}$ adds $\mathcal{O}((k-2)N^3)$ for $k \geq 2$. Given the supports, M patches, and hidden dimension D, applying them to the queries requires

$$\mathcal{O}(BMkN^2 D). \tag{69}$$

Algebraically, identity feature maps require no extra matrix multiplication. The shared query, key, value, and output projections cost $\mathcal{O}(BNMD^2)$, and the feed-forward network costs $\mathcal{O}(BNMDD_{\mathrm{ff}})$, where D_{ff} is its hidden width.

Temporal attention is independently computed for each channel and requires

$$\mathcal{O}(BNM^2 D). \tag{70}$$

Thus, evaluating a block with explicit dense supports and algebraically simplified identity maps costs

$$\mathcal{O}\left(N^2 D_e + (k-2)N^3 + BMkN^2 D + BNM^2 D + BNM(D^2 + DD_{\mathrm{ff}})\right). \tag{71}$$

The adaptive adjacency matrix is shared across samples and patch positions within each layer and is constructed only once per layer in each forward pass.

The current generic implementation additionally constructs an unused embedding-conditioned feature tensor and contracts with dense identity feature maps. These operations can add $\mathcal{O}(ND_e kD^2 + BMNkD^2)$ work even though they introduce no learned feature transformation into the output. Accordingly, the simplified expression describes the model's algebraic computation, not an assertion that these implementation costs are absent.

F PSEUDOCODE

Algorithm 1 summarizes the forecasting computation in Section 3 and the Huber objective used during training. Graph construction occurs once per layer; its weights are shared across the samples and patch positions in that layer. All indexed assignments below are evaluated over their full ranges unless a loop states otherwise.

Algorithm 1 τformer forecasting and training objective

Require: History $\mathbf{X} \in \mathbb{R}^{B\times L\times N}$; learned model parameters;
patch length P, stride S, depth L_e, supports $k \geq 2$,
temperature $\tau > 0$; targets $\mathbf{Y}$ and threshold $\delta > 0$ during training

Ensure: Forecast $\widehat{\mathbf{Y}} \in \mathbb{R}^{B\times T\times N}$; loss $\mathcal{L}$ during training

Compute and detach input-window statistics $\mu_{b,n}$ and $\sigma^2_{b,n}$

$\widetilde{X}_{b,l,n} \leftarrow (X_{b,l,n} - \mu_{b,n})/\sqrt{\sigma^2_{b,n} + \epsilon}$

Replication-pad $\widetilde{\mathbf{X}}$ by S steps and extract patches $\mathbf{p}_{b,n,m} \in \mathbb{R}^P$

$\mathbf{H}^{(0)}_{b,n,m} \leftarrow \mathrm{Dropout}(\mathbf{W}_p\mathbf{p}_{b,n,m} + \mathbf{b}_p + \mathbf{e}^{\mathrm{pos}}_m)$

for $\ell = 1, \ldots, L_e$ **do**

  $(\mathbf{Q}^{(\ell)}, \mathbf{K}^{(\ell)}, \mathbf{V}^{(\ell)}) \leftarrow \mathrm{Proj}^{(\ell)}_{QKV}(\mathbf{H}^{(\ell-1)})$

  $C^{(\ell)}_{ij} \leftarrow \mathrm{ReLU}\left(\frac{\langle \mathbf{e}^{(\ell)}_i, \mathbf{e}^{(\ell)}_j\rangle}{\|\mathbf{e}^{(\ell)}_i\|_2 \|\mathbf{e}^{(\ell)}_j\|_2}\right)$

  $\mathbf{A}^{(\ell)}_\tau \leftarrow \mathrm{Softmax}_{\mathrm{row}}(\mathbf{C}^{(\ell)}/\tau)$

  Construct $\mathbf{G}^{(\ell)}_i = T_i(\mathbf{A}^{(\ell)}_\tau)$, $i = 0, \ldots, k-1$, using Appendix D

  $\widetilde{\mathbf{Q}}^{(\ell)}_{b,:,m,:} \leftarrow \sum_{i=0}^{k-1} \mathbf{G}^{(\ell)}_i \mathbf{Q}^{(\ell)}_{b,:,m,:}$

  Split $\widetilde{\mathbf{Q}}^{(\ell)}$, $\mathbf{K}^{(\ell)}$, and $\mathbf{V}^{(\ell)}$ into attention heads

  for each sample b, channel n, and head h **do**

    $\mathbf{O}^{(\ell,h)}_{b,n} \leftarrow \mathrm{Softmax}\left(\frac{\widetilde{\mathbf{Q}}^{(\ell,h)}_{b,n}(\mathbf{K}^{(\ell,h)}_{b,n})^\top}{\sqrt{D_h}}\right)\mathbf{V}^{(\ell,h)}_{b,n}$

  end for

  Concatenate the head outputs and apply the attention output projection

  Apply residual, normalization, and feed-forward operations to obtain $\mathbf{H}^{(\ell)}$

end for

$\widehat{\widetilde{\mathbf{y}}}_{b,n} \leftarrow \mathbf{W}_o\,\mathrm{vec}(\mathbf{H}^{(L_e)}_{b,n,:,:}) + \mathbf{b}_o$

$\widehat{Y}_{b,t,n} \leftarrow \sqrt{\sigma^2_{b,n} + \epsilon}\,\widehat{\widetilde{Y}}_{b,t,n} + \mu_{b,n}$

if training **then**

  $\mathcal{L} \leftarrow \frac{1}{BTN}\sum_{b=1}^{B}\sum_{t=1}^{T}\sum_{n=1}^{N}\ell_\delta(\widehat{Y}_{b,t,n} - Y_{b,t,n})$

  return $\widehat{\mathbf{Y}}$ and $\mathcal{L}$ for backpropagation

end if

return $\widehat{\mathbf{Y}}$

The attention softmax is taken over key patches, and keys and values are projected from the current block input. Dropout and the standard block operations follow Appendix D; attention-weight dropout is applied during training. De-normalization restores the scale of the model input. For the experiments, this is the scale obtained using training-set standardization.

For post-hoc analysis, a detached base adjacency $\mathbf{A}$ is used to compute $\widehat{\alpha} = (\mathrm{tr}(\mathbf{A}) - 1)/(N-1)$, $\mathbf{R} = \mathbf{A} - \widehat{\alpha}\mathbf{I} - (1-\widehat{\alpha})\mathbf{U}$, and $\mathcal{S}(\mathbf{A}) = \|\mathbf{R}\|^2_F/(N-1)$. These diagnostics are separate from the training objective.

G NOTATION

Table 3 summarizes the notation in Section 3. The final group adds the normalization and patching quantities used in Algorithm 1. Tensor shapes include the batch dimension; the problem definition omits it for a single sequence.

Indexing conventions. Superscripts (ℓ) and (ℓ, h) identify an encoder layer and a layer–head pair, respectively. The layer index is suppressed when discussing a single channel graph. A colon retains an entire tensor axis: for example, $\mathbf{Q}^{(\ell)}_{b,:,m,:} \in \mathbb{R}^{N\times D}$ and $\mathbf{H}^{(L_e)}_{b,n,:,:} \in \mathbb{R}^{M\times D}$. For queries, keys, and values, selecting (b, n, ℓ, h) yields a sequence in $\mathbb{R}^{M\times D_h}$. Nonbold indexed entries, such as $Y_{b,t,n}$, are scalars. Letter labels such as $p, o, Q, K, V, \mathrm{pos}$ identify roles; they are explained with the corresponding quantities below.

Table 3: Symbols, index ranges, and tensor dimensions.

Symbol	Definition
Indices and dimensions	
B, b	Batch size; sample index $b = 1, \ldots, B$.
L, l	History length; historical time index $l = 1, \ldots, L$.
T, t	Forecast horizon; future time index $t = 1, \ldots, T$, corresponding to absolute position $L + t$.
N, n, i, j, r	Number of channels; channel indices in $\{1, \ldots, N\}$. In graph entries, i receives from j; r indexes the softmax sum.
P, M, m	Patch length, number of patches per channel, and patch index $m = 1, \ldots, M$.
L_e, ℓ	Encoder depth (e: encoder); layer index $\ell = 1, \ldots, L_e$, with 0 denoting the encoder input.
D, D_e, D_h, h	Hidden width, channel-embedding width (e: embedding), and width per attention head; head index $h = 1, \ldots, D/D_h$.
k, i	Number of graph supports; polynomial order $i = 0, \ldots, k-1$. The index i denotes a channel in graph-entry formulas.
Sequences, patch representations, and forecasting	
$\mathbf{x}_l, \mathbf{x}_{L+t}$	Observed and future time-step vectors, respectively; each lies in $\mathbb{R}^N$.
$\mathbf{X}, \mathbf{Y}$	Historical input and future target tensors; shapes $B \times L \times N$ and $B \times T \times N$, respectively.
$\widehat{\mathbf{Y}}$	Forecast after de-normalization; shape $B \times T \times N$.
$\widehat{\widetilde{\mathbf{y}}}_{b,n}$	Forecast for sample b and channel n on the normalized scale; a vector in $\mathbb{R}^T$, whose t-th entry is $\widehat{\widetilde{Y}}_{b,t,n}$.
f, f_θ	Forecasting map $\mathbb{R}^{L\times N} \to \mathbb{R}^{T\times N}$; θ denotes the model parameters when written explicitly.
$\mathbf{p}_{b,n,m}$	Normalized input patch; a vector in $\mathbb{R}^P$.
$\mathbf{W}_p, \mathbf{b}_p$	Patch-projection weight and bias (p: patch); shapes $D \times P$ and D.
$\mathbf{e}^{\mathrm{pos}}_m$	Learnable patch-position embedding (pos: position); a vector in $\mathbb{R}^D$.
$\mathbf{H}^{(\ell)}$	Patch representations after layer ℓ; shape $B \times N \times M \times D$. $\mathbf{H}^{(0)}$ denotes the initial patch embeddings.
$\mathbf{W}_o, \mathbf{b}_o$	Forecasting-head weight and bias (o: forecast output); shapes $T \times MD$ and T.

Continued on the next page

Table 3 (continued)

Symbol	Definition
Channel graph and structural metrics	
$\mathbf{E}^{(\ell)}$, $\mathbf{e}_i$	Learnable channel-embedding matrix and its channel-i vector; shapes $N \times D_e$ and D_e.
$\mathbf{C}$, C_{ij}	Non-negative cosine-similarity matrix and its (i, j) entry; shape $N \times N$.
τ	Positive graph-softmax temperature.
$\mathbf{A}_\tau^{(\ell)}$, $A_{ij}^{(\tau)}$	Layer-specific row-stochastic channel adjacency and an entry with the layer index omitted; shape $N \times N$. $A_{\tau,ij}$ is the same entry; $\mathbf{A}$ suppresses both layer and temperature labels.
$\mathbf{I}$, $\mathbf{U}$	Identity and uniform-mixing matrices; $\mathbf{U} = \mathbf{1}\mathbf{1}^\top/N$, both of shape $N \times N$.
$\mathbf{1}$	All-ones vector in $\mathbb{R}^N$.
α, $\widehat{\alpha}$	Candidate identity–uniform coefficient and its projection value (Identity Strength), $\widehat{\alpha} = (\mathrm{tr}(\mathbf{A}) - 1)/(N - 1)$.
$\mathbf{R}(\alpha)$, $\mathbf{R}$	Graph residual $\mathbf{A} - \alpha\mathbf{I} - (1 - \alpha)\mathbf{U}$; $\mathbf{R} = \mathbf{R}(\widehat{\alpha})$, of shape $N \times N$.
$\mathcal{S}(\mathbf{A})$	Interaction Heterogeneity $\|\mathbf{R}\|_F^2/(N - 1)$.
Graph propagation, temporal attention, and loss	
$\mathbf{W}_Q^{(\ell)}$, $\mathbf{W}_K^{(\ell)}$, $\mathbf{W}_V^{(\ell)}$	Query, key, and value projection matrices (Q, K, V: projection roles); each has shape $D \times D$.
$\mathbf{Q}^{(\ell)}$, $\mathbf{K}^{(\ell)}$, $\mathbf{V}^{(\ell)}$	Query, key, and value tensors; each has shape $B \times N \times M \times D$.
$\widetilde{\mathbf{Q}}^{(\ell)}$	Query tensor after channel-graph propagation; shape $B \times N \times M \times D$.
$T_i(\cdot)$, $\mathbf{G}_i^{(\ell)}$	Order-i Chebyshev polynomial and graph support $\mathbf{G}_i^{(\ell)} = T_i(\mathbf{A}_\tau^{(\ell)})$; the support has shape $N \times N$.
$\mathbf{O}_{b,n}^{(\ell,h)}$	Output of one temporal-attention head; shape $M \times D_h$.
$\ell(\cdot)$, $\ell_\delta(\cdot)$, δ	Scalar Huber penalty; $\ell(\cdot)$ omits the positive threshold δ made explicit in $\ell_\delta(\cdot)$.
$\mathcal{L}$	Huber loss averaged over samples, future times, and channels.
Additional quantities in the pseudocode	
$\mu_{b,n}$, $\sigma_{b,n}^2$	Input-window mean and variance for each sample and channel.
ϵ	Positive stabilizer in the normalization denominator.
$\widetilde{\mathbf{X}}$	Input after per-sample, per-channel temporal normalization; shape $B \times L \times N$.
S	Patch stride, also the right-padding length.

Operators and labels. $\mathrm{Proj}_{QKV}^{(\ell)}$ collects the three feature projections; QKV is a role label. Graph constructs the channel adjacency, GProp propagates queries over graph supports, and $\mathrm{AttnBlock}$ denotes temporal attention with residual, normalization, and feed-forward operations. Graph softmax acts over source channels; attention softmax acts over key patches. tr and vec denote trace and vectorization. The brackets $\langle \cdot, \cdot \rangle$ denote the Euclidean inner product. The superscript $\top$ denotes transpose, the subscript 2 denotes the Euclidean norm, and F denotes the Frobenius norm or inner product in $\| \cdot \|_F$ and $\langle \cdot, \cdot \rangle_F$. Parenthesized layer/head indices and the temperature label in $A_{ij}^{(\tau)}$ are not powers.

H TEST-ONLY GAUSSIAN NOISE EVALUATION

Noise injection has also been used to assess forecasting robustness in FiLM (Zhou et al., 2022, Appendix G). For Figure 6, we evaluate τformer, FiLM, PCD, and MSGNet on ETTh2 using an input length of 96. For each model and prediction horizon, we select a checkpoint by its lowest loss on clean validation data. The same checkpoint is evaluated at every noise level without retraining or fine-tuning; the clean point at $\sigma = 0$ is measured with that checkpoint rather than taken from the main forecasting table. The models retain their respective training settings, so the shared test perturbations do not imply identical optimization protocols. For PCD, the correlation matrix computed from clean training data remains fixed during noisy test evaluation.

Each channel is standardized using its mean and standard deviation from the clean training period. Before any model-internal normalization, Gaussian noise is added only to the standardized observations at time positions covered by the test input windows, including their historical context. For channel n and time position t, the corrupted value is $x_{t,n}^{\text{noisy}} = x_{t,n} + \varepsilon_{t,n}$, where $\varepsilon_{t,n} \sim \mathcal{N}(0, \sigma^2)$ is sampled independently across time positions and channels. Thus, σ is measured in units of each channel's clean-training standard deviation. Figure 6 displays $\sigma \in \{0, 0.1, 0.2, 0.3, 0.4\}$.

For each noise level, corruption is generated once on the standardized input timeline before extracting overlapping windows of length 96. A shared observation therefore receives the same perturbation in every window that contains it. Future targets and time markers remain clean. The plotted value at each noise level is the arithmetic mean of the four MAEs for prediction horizons 96, 192, 336, and 720.

I FUTURE WORK

Building on this work, future research could develop criteria for selecting channel strategies based on data characteristics. More broadly, exploring channel strategies in more complex forecasting frameworks, including agentic systems, MoE architectures, and diffusion models, could lead to a unified mechanism for regulating channel interaction across them.