Persistent Structure Meets Dynamic Attention: Cross-Variable Priors for Multivariate Time Series Forecasting

Chinmoy Mohapatra
cmohapat@amazon.com
Amazon
Seattle, WA, United States

Suyash Vishnoi
suyashv@amazon.com
Amazon
Seattle, WA, United States

Abhislasha Katariya
abhkata@amazon.com
Amazon
Seattle, WA, United States

Rohit Malshe
malshe@amazon.com
Amazon
Seattle, WA, United States

Julian Pachon
pachonj@amazon.com
Amazon
Seattle, WA, United States

Abstract

Multivariate time series contain two kinds of cross-variable relationships: persistent ones that reflect underlying structure (geographic proximity, shared infrastructure, physical coupling) and dynamic ones that arise from transient conditions in each observation window. Current transformer architectures conflate the two—channel-independent models ignore cross-variable relationships entirely, while cross-variable methods must rediscover all structure from scratch in every forward pass. We propose *Factored Topology Bias* (FTB), which explicitly separates these concerns by adding a learned low-rank prior $PP^\top$ to cross-variable attention logits, encoding time-invariant relationships, while standard attention ($QK^\top$) captures input-specific dynamics. The factored form requires only $O(Cd_p)$ parameters and is interleaved with temporal attention at every encoder layer, controlled by a learned gate. On standard benchmarks spanning 21–862 variables, FTB improves forecasting accuracy by 2–7% on high-dimensional datasets while correctly providing no benefit when persistent structure is absent. Ablation reveals a horizon-dependent complementarity: the structural prior contributes 33% of the gain at short horizons but 88% at long horizons, confirming that time-invariant knowledge compensates as temporal signal fades. We introduce a *Params-Per-Pair* diagnostic that predicts from dataset properties alone whether structural priors will help.

CCS Concepts

• **Computing methodologies** → **Neural networks**; *Learning latent representations*.

MILETS '26, Jeju, Korea

https://doi.org/10.1145/nnnnnnn.nnnnnnn

Keywords

time series forecasting, multivariate, cross-variable attention, structural priors, transformer

ACM Reference Format:
Chinmoy Mohapatra, Suyash Vishnoi, Abhislasha Katariya, Rohit Malshe, and Julian Pachon. 2026. Persistent Structure Meets Dynamic Attention: Cross-Variable Priors for Multivariate Time Series Forecasting. In *The 12th Mining and Learning from Time Series Workshop (MILETS '26), August 3–4, 2026, Jeju, Korea*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Multivariate time series forecasting underpins applications from energy grid management to traffic control and climate monitoring. As the number of monitored variables C grows into the hundreds or thousands, the number of pairwise relationships grows quadratically ($C(C-1)/2$), and many of these relationships are *persistent*: co-located sensors always co-vary, customers in the same district share load patterns, adjacent highway segments experience correlated congestion. Yet current transformer architectures either ignore this structure entirely or must rediscover it from scratch in every forward pass.

Transformer-based methods have become competitive but face a fundamental architectural tradeoff. Channel-independent (CI) models such as PatchTST [5] process each variable through shared encoder weights, effectively multiplying training data by the number of channels C. This yields strong regularization but discards all cross-variable dependencies. At the other extreme, cross-variable methods add explicit inter-variable mixing: iTransformer [4] treats variables as tokens, TSMixer [1] alternates temporal and channel MLP layers, and Crossformer [17] computes full $O(C^2)$ cross-variable attention. None of these methods distinguish *persistent* structural relationships (co-located sensors, same customer segment) from *dynamic* input-dependent correlations.

Consider a CI model forecasting electricity demand for 321 clients independently: it cannot exploit the fact that two clients in the same residential neighborhood always co-move, because no information flows between their representations.

Cross-variable methods (TSMixer, iTransformer) can in principle discover such relationships, but they treat a persistent geographic co-variation identically to a transient correlation that appears in one batch and vanishes in the next. Current architectures lack mechanisms to separate time-invariant from time-varying cross-variable relationships.

We propose *Factored Topology Bias* (FTB), which separates these two types of cross-variable knowledge architecturally (Figure 1). We learn an embedding $P \in \mathbb{R}^{C \times d_p}$ and add the bias $B = PP^\top$ to cross-variable attention logits. This creates an explicit decomposition: the dynamic term $QK^\top$ (recomputed per input) captures transient, input-dependent relationships, while the persistent term $PP^\top$ (learned once, shared across all inputs) encodes which variables are structurally related regardless of the current observation. FTB interleaves this augmented cross-variable attention with temporal attention at every encoder layer, controlled by a learned gate: at g=0 the model recovers PatchTST exactly, biasing the architecture toward CI behavior unless cross-variable information demonstrably reduces loss.

The topology prior requires only $O(Cd_p)$ parameters (1,926 for Electricity, <0.2% of the model), while the full cross-variable pathway (Q/K/V projections, FFN, layer norms across 3 layers) adds ∼25% overhead—comparable to TSMixer's channel mixing (∼164K across 8 layers). FTB's contribution is making this cross-variable investment pay off at long horizons where alternatives (TSMixer, CrossOnly attention without topology) fail to generalize.

Our contributions are as follows:

(1) We propose an interleaved gated architecture that augments CI transformers with cross-variable attention while preserving CI regularization. On Electricity (321 variables), FTB improves MSE by 2–7% across all horizons and outperforms TSMixer at 3 of 4 horizons. On Traffic (862 variables), improvement is 2.5%.
(2) We introduce a *Params-Per-Pair* (PPP) diagnostic that predicts from dataset dimensions alone whether structural priors will help. FTB improves when PPP < 40 ($C > 100$); on Weather (C=21, PPP = 942), it correctly provides no benefit.
(3) We demonstrate via ablation that the topology prior's contribution shifts from 33% at short horizons to 88% at long horizons, establishing that time-invariant relationships become critical as prediction difficulty grows.
(4) We show that without $PP^\top$, cross-variable attention alone barely outperforms CI (−0.9% at H=336). The structural prior is what makes cross-variable attention generalize rather than overfit.

2 Related Work

Cross-variable modeling in time series. Channel-independent architectures [5, 16] trade cross-variable information for regularization. A key finding of [5] was that this simple approach often outperforms explicit cross-variable methods, underscoring how easily cross-variable modeling overfits. Subsequent work has pursued better-constrained alternatives:

Table 1: Design comparison with cross-variable methods.

	FTB	TSMixer	iTransf.	GraphWN
Cross-var mechanism	Attn+bias	MLP	Attn	GNN
Persistent prior	$PP^\top$	None	None	Adj. A
Temporal preserved	✓	✓	×	✓
CI regularization	✓	×	×	×
Params scale w/ C	$O(Cd_p)$	$O(Cd)$	$O(d^2)$	$O(Cd)$
Plug-in to CI arch.	✓	×	×	×

iTransformer [4] treats variables as tokens (full cross-variable attention without CI regularization), TSMixer [1] alternates temporal and channel MLP layers (rank-bottlenecked mixing), Crossformer [17] uses cross-dimension attention with learned routing, CARD [9] blends CI and cross-variable predictions via learned channel alignment, TimeXer [10] incorporates exogenous variables through cross-attention, and graph-based methods [13, 14] learn adjacency matrices for GNN message passing. Relative to all of these, FTB preserves CI regularization while adding a learned structural prior as an attention bias, explicitly separating time-invariant from dynamic relationships.

Structural biases in attention. Adding persistent, input-independent biases to attention logits originates in positional encoding: sinusoidal PE [8], relative PE [7], and ALiBi [6] all encode fixed positional relationships as additive terms. Graphormer [15] extends this from sequential positions to graph topology using spatial and edge encodings. In time series, Moirai [11] adds a binary same/different-variate bias, and GOAT [3] derives from Entropic Optimal Transport that the optimal attention prior takes the form $\mathrm{softmax}(s + \log \pi)$, parameterizing temporal priors with Fourier features. FTB applies this log-prior principle to the *variable* dimension rather than the temporal dimension, with a factored PSD form that provides $O(Cd_p)$ parameter efficiency and implicit spectral regularization through the low-rank constraint. Unlike Graphormer (which requires a pre-defined graph and encodes fixed shortest-path distances), FTB learns its topology end-to-end from forecasting loss, discovering task-relevant structure that may differ from raw statistical correlation or domain graphs.

3 Method

Given a multivariate time series $X \in \mathbb{R}^{L \times C}$ with L lookback steps and C variables, we predict $\hat{Y} \in \mathbb{R}^{H \times C}$ for H future steps. Our backbone is PatchTST [5], which divides the lookback into N overlapping patches per variable (patch length 16, stride 8), embeds them into d-dimensional representations, and applies L_{enc} layers of temporal multi-head self-attention independently per variable. Because each variable is processed by identical shared weights, the effective training set is multiplied by C, providing strong regularization at the cost of preventing any cross-variable information flow.

FTB extends this backbone with cross-variable attention augmented by a persistent structural prior. We achieve this through three design choices: (1) a factored topology bias that encodes persistent relationships cheaply, (2) an interleaved

architecture that injects this knowledge throughout learning, and (3) a gated fusion that allows safe fallback to CI behavior when cross-variable information does not help.

3.1 Factored Topology Bias

We learn a topology embedding $P \in \mathbb{R}^{C \times d_p}$ where each row p_i represents variable i's structural identity in a low-dimensional latent space ($d_p \ll C$). The topology bias matrix is:

$$B = PP^\top \in \mathbb{R}^{C \times C} \tag{1}$$

Entry $B_{ij} = \langle p_i, p_j \rangle$ measures the learned persistent affinity between variables i and j: variables with similar embeddings (same cluster, same geographic region, similar consumption pattern) receive high affinity. This construction is analogous to collaborative filtering in recommender systems, where each item receives a short embedding and affinity is their inner product.

The factored form $PP^\top$ guarantees three properties by construction:

- **Symmetry** ($B_{ij} = B_{ji}$): if variable i is structurally related to j, the reverse holds. This is appropriate for undirected relationships such as geographic proximity or shared load patterns.
- **Low-rank** (rank $\leq d_p$): the model must explain all $C(C-1)/2$ pairwise relationships through only d_p latent factors, providing implicit regularization [2]. Gradient descent on this factored form converges toward minimum nuclear norm solutions, equivalent to implicit weight decay on the embeddings.
- **Parameter efficiency** ($O(Cd_p)$ vs $O(C^2)$): for Electricity (C=321, d_p=6), the topology requires 1,926 parameters versus 103,041 for a full unconstrained matrix, a 54× reduction that forces meaningful compression.

Cross-variable attention at each patch position applies:

$$A = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + \lambda \cdot PP^\top\right), \quad H_{\text{cross}} = AV \tag{2}$$

where Q, K, V project over the variable dimension and λ is a learnable per-head scalar. The dynamic term $QK^\top$ captures input-specific correlations that may differ from batch to batch, while the persistent term $PP^\top$ encodes structural affinities that hold regardless of the current input. Both contribute through softmax normalization, with their relative influence governed by the learned λ.

Connection to optimal transport. GOAT [3] derives from Entropic Optimal Transport that optimal attention takes the form $\text{softmax}(s + \log \pi)$ where π is a learned prior. Our $\lambda \cdot PP^\top$ serves as the log-prior for the variable dimension. Unlike GOAT's temporal Fourier parameterization, which encodes shift-equivariant patterns, $PP^\top$ captures *topological* structure (clusters, hierarchies) with no equivariance constraint, which is appropriate since variable relationships have no inherent ordering.

Algorithm 1 FTB Encoder Layer

Require: $H \in \mathbb{R}^{B \times C \times N \times d}$, topology $P \in \mathbb{R}^{C \times d_p}$
Ensure: $H' \in \mathbb{R}^{B \times C \times N \times d}$

```
H_temp ← TemporalMHA(H)                          ▷ Per-variable, over N patches
for each patch position t = 1, ..., N do
    Q, K, V ← H_temp[:, :, t, :] · W_Q, W_K, W_V
    L ← QK^⊤/√d_k + λ · PP^⊤                     ▷ Dynamic + persistent
    Δ[:, :, t, :] ← FFN(LN(H_temp[:, :, t, :] + softmax(L)VW_O))
end for
g ← σ(w_g)                                       ▷ Learned gate, init ≈ 0.27
H' ← LN(H + H_temp + g · Δ)                      ▷ Gated residual
```

3.2 Interleaved Gated Architecture

Each encoder layer (Figure 2) performs three operations in sequence:

(1) **Temporal self-attention**: standard multi-head attention over patches for each variable independently (identical to PatchTST), preserving CI data efficiency through shared weights.
(2) **Cross-feature attention with $PP^\top$**: multi-head attention across variables at the same patch position, with additive topology bias (Eq. 2). Given representations $H_t \in \mathbb{R}^{B \times C \times d}$ at patch t, we project $Q{=}H_tW_Q$, $K{=}H_tW_K$, $V{=}H_tW_V$ and attend over the C dimension.
(3) **Gated fusion**: the cross-feature output blends back via a learned gate:

$$H' = H_{\text{temp}} + g \cdot \Delta H_{\text{cross}}, \quad g = \sigma(w_g) \tag{3}$$

where w_g is initialized at -1.0 giving $g_0 \approx 0.27$.

Design rationale. The conservative gate initialization ensures FTB starts near CI behavior and only incorporates cross-variable information when gradient signal indicates benefit. If cross-variable mixing is unhelpful, the gate remains low, limiting (though not eliminating) degradation. In practice, the gate converges to $g{\approx}0.53$ on Electricity (substantial cross-variable signal) but barely moves from initialization on Weather ($g{\approx}0.28$), reflecting limited cross-variable value. On Weather, mild degradation (+0.6%) still occurs because the cross-attention pathway itself adds overparameterization that the gate cannot fully suppress; the PPP diagnostic (Section 3.3) provides a more reliable pre-training safeguard.

Interleaving is critical: injecting topology only after the encoder (post-hoc) yields negligible improvement (+0.3%), while interleaving at every layer yields the full 2–7% benefit. Structural information must participate throughout representation learning, not merely correct it afterward.

Computational cost. FTB adds 3 cross-attention layers (∼392K parameters, ∼25% over PatchTST's 1.56M baseline), comparable to TSMixer's channel mixing (∼164K across 8 layers). The topology prior $PP^\top$ itself adds only 1,926 parameters for Electricity (0.12% of baseline). In wall-clock time, FTB trains 1.3–1.5× slower per epoch but converges in fewer epochs, yielding comparable total training time.

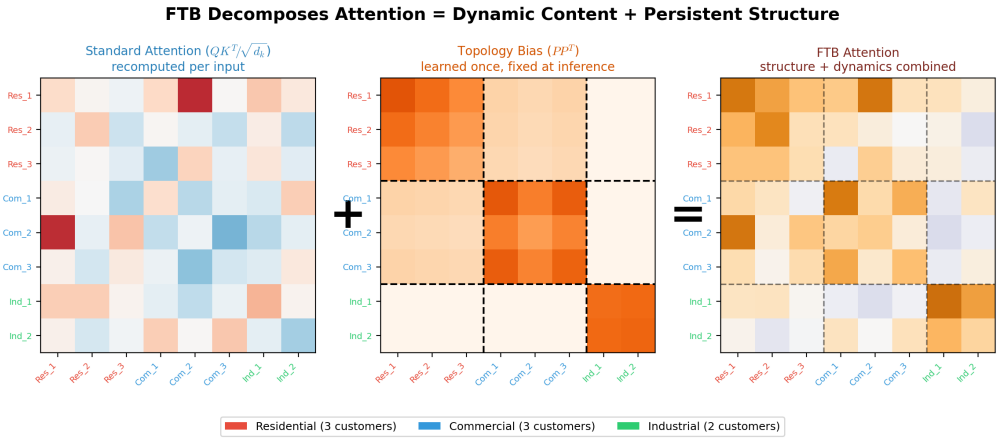

Figure 1: FTB decomposes cross-variable attention into persistent structural affinity ($PP^\top$, learned once) and dynamic content affinity ($QK^\top$, recomputed per input). Example: 8 electricity customers in 3 groups.

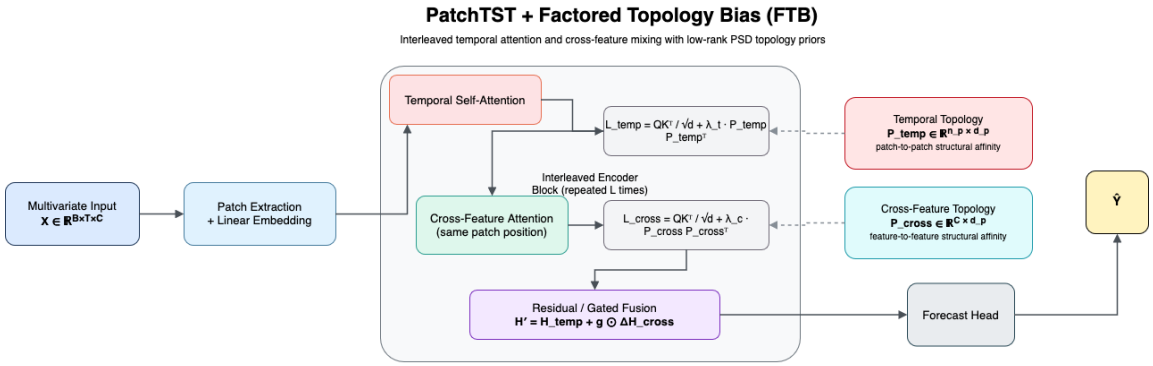

Figure 2: FTB-PatchTST architecture. Each encoder layer interleaves temporal self-attention (per variable) with cross-feature attention (per patch position), augmented by $PP^\top$. A learned gate controls cross-variable contribution.

3.3 When Does It Help? The PPP Diagnostic

Not all datasets benefit from cross-variable attention. We introduce the *Params-Per-Pair* (PPP) ratio:

$$\text{PPP} = \frac{\theta_{\text{cross}}}{C(C-1)/2} \tag{4}$$

where θ_{cross} is the fixed cross-attention parameter count and $C(C-1)/2$ counts unique variable pairs. PPP measures parameters available per pairwise relationship.

When PPP $\gg 1$ (Weather: 198K params / 210 pairs = 942), cross-attention has vastly more capacity than relationships to model, enabling memorization of spurious correlations including noise. When PPP $\approx$ 1–10 (Electricity: 198K / 51K pairs = 3.9), the compression bottleneck forces the model to learn shared, generalizable patterns; only the dominant structural modes fit. The topology prior $PP^\top$ is most valuable in this regime: it provides the structural scaffold that guides compression toward genuine persistent patterns rather than transient correlations.

Empirically, FTB helps reliably when PPP < 40 (i.e., $C > 100$ with our 3-layer architecture). The threshold is validated across three datasets: Weather (C=21, PPP = 942, no benefit), Electricity (C=321, PPP = 3.9, -2.1% to -7.3%), and Traffic (C=862, PPP = 0.53, -2.5%). For a new dataset, the decision requires only computing C; no training is needed.

4 Experiments

4.1 Setup

Datasets. **Electricity** (321 clients, hourly, ∼26K steps), **Traffic** (862 road sensors, hourly, ∼17K steps), **Weather** (21 meteorological variables, 10-min, ∼52K steps).

Baselines. PatchTST [5] (CI transformer, our backbone), TSMixer [1] (MLP mixing), TimesNet [12] (CNN-based), DLinear [16] (linear), FEDformer [18] (frequency attention), iTransformer [4] (variables as tokens).

Protocol. We use lookback = 336 and prediction horizons $\{96, 192, 336, 720\}$, following [5]. FTB uses d_p=6, 3 cross-attention layers, d_{model}=128. All baselines use their published default settings.

Table 2: MSE on Electricity (321 features), lookback=336. Best in bold, second best underlined. †Published values.

Model	96	192	336	720	Avg
DLinear†	0.197	0.196	0.209	0.245	0.212
FEDformer†	0.193	0.201	0.214	0.246	0.214
TimesNet	0.186	0.194	0.205	0.227	0.203
iTransformer	0.154	0.175	0.196	0.235	0.190
PatchTST	0.159	0.178	0.194	0.235	0.191
TSMixer	**0.150**	0.174	0.190	0.228	0.185
FTB (Ours)	0.155	**0.167**	**0.184**	**0.225**	**0.183**

All experiments use the Time-Series-Library codebase with identical train/val/test splits (70/10/20 chronological). Multi-seed validation (seeds 2021, 42, 7) on Electricity at lb=96 yields: PatchTST 0.2117±0.0003, FTB 0.1716±0.0009, CrossOnly 0.179±0.001 (mean ± std over 3 seeds). The worst FTB seed (0.1722) beats the best PatchTST seed (0.2115). Our reported improvements of 2–7% at lb=336 are 10–35× larger than seed-level variance.

4.2 Main Results

Table 2 presents results on Electricity at the standard lookback-336 protocol using FTB's default configuration (d_p=6), which improves consistently across all horizons; Section 4.3 shows d_p=16 yields larger gains at short horizons at the cost of overfitting at H=720. FTB achieves the best average MSE (0.183) across all seven baselines, outperforming TSMixer (the strongest baseline) at prediction horizons 192, 336, and 720. Margins over older methods are substantial: 8–22% over DLinear, 8–17% over FEDformer, and 1–17% over TimesNet.

The iTransformer comparison is particularly informative. At matched model capacity (d_{model}=128), iTransformer is competitive at H=96 (0.154 vs FTB's 0.155) but degrades at longer horizons: at H=336, it is *worse than PatchTST* (0.196 vs 0.194) despite having full cross-variable attention. Without a structural prior to distinguish persistent from transient relationships, unconstrained cross-variable attention overfits at long horizons—precisely the failure mode that $PP^\top$ addresses. FTB beats iTransformer by 4.6% at H=192 and 6.5% at H=336.

The results reveal a horizon-dependent ordering among cross-variable methods. TSMixer wins at H=96 (+3.3% over FTB), where recent temporal patterns suffice for rank-32 MLP compression. FTB dominates at longer horizons: −3.8% at H=192, −3.2% at H=336, and −1.3% at H=720 over TSMixer. This crossover previews our central mechanistic finding (Section 4.3): persistent structural knowledge becomes most valuable when temporal patterns begin to fade.

Traffic (C=862). PatchTST achieves 0.405 MSE and FTB achieves 0.395 (−2.5%). Per-sensor analysis shows that 95% of 862 sensors improve individually, with 2× differential improvement across learned clusters. With PPP = 0.53 (198K parameters for 371K unique sensor pairs), the cross-attention is strongly undercapacitated and forced to discover only the most dominant structural patterns.

Table 3: Ablation on Electricity, lb=336. CrossOnly = interleaved gated architecture without $PP^\top$. PP$^\top$ share = fraction of total gain attributable to topology.

	pl=96		pl=336	
Configuration	MSE	Δ%	MSE	Δ%
PatchTST (baseline)	.1585	—	.1940	—
+ CrossOnly (no $PP^\top$)	.1542	−2.7	.1923	−0.9
+ FTB d_p=6	.1552	−2.1	.1839	−5.2
+ FTB d_p=16	**.1522**	−4.0	**.1798**	−7.3
$PP^\top$ share (d_p=16):	33%		88%	

Weather (C=21). FTB provides no benefit (+0.6% degradation at H=96). With only 21 variables producing 210 unique pairs, PPP = 942: each pair has nearly 1,000 parameters available, enabling memorization of spurious correlations including temporal noise. The cross-attention pathway overfits without the compression bottleneck that makes structural priors valuable. This negative result *validates* the PPP diagnostic rather than indicating method failure; it correctly identifies the regime where cross-variable attention should not be applied without heavy regularization. Reducing to 1 cross-attention layer with weight decay restores non-degradation on this dataset.

4.3 Horizon-Dependent Complementarity

Table 3 presents the central mechanistic finding of this paper. The CrossOnly variant (our full interleaved gated architecture *without* $PP^\top$) shows markedly horizon-dependent behavior:

- **At pl=96**: CrossOnly (−2.7%) provides two-thirds of the FTB gain. Dynamic attention discovers most cross-variable relationships when temporal patterns are strong; the topology prior adds a useful but non-dominant 33%.
- **At pl=336**: CrossOnly degrades to just −0.9%, while FTB achieves −7.3%. The topology prior $PP^\top$ provides **88% of the total gain**. As temporal signal fades, persistent structural knowledge becomes the dominant source of improvement.

This decomposition establishes a specific mechanism: the structural prior and dynamic attention are complementary signals whose relative importance shifts with prediction difficulty. At short horizons, recent temporal context is highly predictive. The $QK^\top$ term discovers relevant cross-variable relationships in each forward pass because temporal patterns clearly indicate which variables currently co-move. At long horizons, temporal patterns fade and dynamic attention produces noisier estimates across the $C \times C$ relationship space. The structural prior $PP^\top$ compensates because it encodes relationships that hold *regardless of input*: a structural guarantee rather than an input-dependent inference.

Table 4: Within-dataset scaling: Electricity subsampling at lb=336, pl=96.

C	PPP	PatchTST	FTB	Δ%
50	161	0.1570	0.1560	−0.7
100	40	0.1495	0.1484	−0.8
200	10	0.1435	0.1417	−1.3
321	3.9	0.1585	0.1552	−2.1

This mechanism also explains FTB's advantage over TSMixer at long horizons (Table 2). TSMixer's MLP bottleneck performs implicit rank-32 compression of all cross-variable information without distinguishing persistent from dynamic relationships. At short horizons this suffices because recent temporal patterns carry most information. At long horizons the persistent/dynamic separation becomes critical, and FTB's explicit encoding of persistent structure via $PP^\top$ provides what rank-32 MLP compression cannot.

Supporting evidence. Two additional findings rule out alternative explanations.

First, post-encoder injection of $PP^\top$ (adding the topology bias only to a final cross-attention layer after PatchTST's encoder) yields only −0.3%. The structural prior must participate *throughout* representation learning, shaping how temporal features are built layer by layer, rather than merely correcting the final representation.

Second, applying FTB to iTransformer (which already has native, unconstrained cross-variable attention) yields no improvement (∼0% vs iTransformer alone). This confirms that $PP^\top$ encodes information that unconstrained cross-variable attention discovers on its own. FTB's value lies in providing this information to architectures that lack native cross-variable access. It bridges the gap between CI regularization and cross-variable knowledge, rather than being a universal attention improvement.

Topology rank. Our main results use the conservative d_p=6, which improves consistently across all horizons and datasets. Higher rank (d_p=16 ≈ $\sqrt{C}$) amplifies short-horizon gains but overfits at H=720 (−0.3% vs −4.2% for d_p=6). On Traffic (C=862), d_p=6 remains optimal; higher rank progressively diminishes improvement. This divergence reflects structural heterogeneity: Electricity has multi-scale sub-clusters (CoV of neighbor counts = 0.55), while Traffic has uniform block structure (CoV = 0.28) where rank-6 saturates the persistent modes.

4.4 Dimensionality Scaling

A cross-dataset comparison (Weather vs Electricity vs Traffic) conflates dimensionality with domain, sampling rate, and distribution properties. To isolate the pure effect of dimensionality, we subsample the Electricity dataset to $C \in \{50, 100, 200, 321\}$ features at exactly matched settings (lb=336, pl=96). The same training protocol, architecture, and hyperparameters are used for each subset; only the number of variables changes.

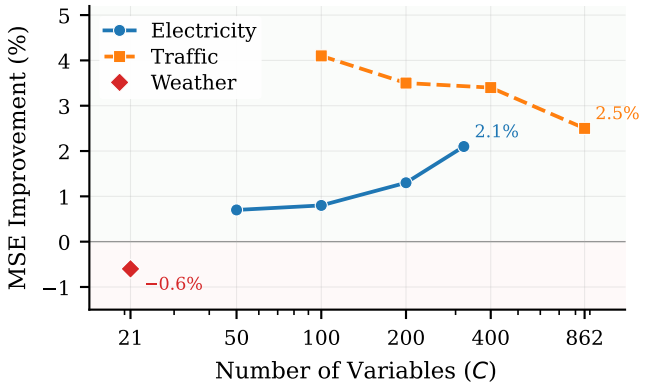

Figure 3: FTB improvement scales with dimensionality. Electricity subsampling shows monotonic increase from 0.7% (C=50) to 2.1% (C=321). Weather shows no benefit (C=21, PPP=942).

Table 5: Traffic subsampling at lb=336, pl=96.

C	PPP	PatchTST	FTB	Δ%
100	40	0.3016	0.2894	−4.1
200	10	0.3033	0.2927	−3.5
400	2.5	0.3186	0.3077	−3.4
862	0.53	0.4050	0.3949	−2.5

Table 4 and Figure 3 show monotonically increasing improvement as C grows. Traffic subsampling (Table 5) confirms consistent improvement of 2.5–4.1% across all sizes, though the trend is flat rather than monotonic. This is explained by d_p=6 undercapacitation: a fixed rank-6 topology covers fewer structural modes as C grows.

The scaling behavior connects directly to the PPP compression argument from Section 3.3. At C=50 (PPP = 161), the cross-attention has 161 parameters available per variable pair, enough to partially memorize noise and spurious batch-specific correlations, limiting the marginal value of a structural prior. At C=321 (PPP = 3.9), the same 198K parameters must compress 51,360 unique pairwise relationships into generalizable patterns. In this strongly undercapacitated regime, the topology prior $PP^\top$ provides critical guidance: it biases the cross-attention toward persistent structural patterns from the first forward pass, rather than requiring the model to discover all pairwise relationships from limited per-pair training signal alone.

The combined evidence across both datasets validates that FTB's benefit is governed by the ratio of structural task complexity ($C(C-1)/2$ pairs) to model capacity (θ_{cross} parameters). Because this ratio is computable from dataset dimensions alone, a practitioner can determine whether FTB applies to their problem before running any experiment.

4.5 What Does the Topology Learn?

The topology embedding P is learned end-to-end from forecasting loss; no supervision on variable relationships is provided. We analyze the learned structure by clustering variables using their P embeddings.

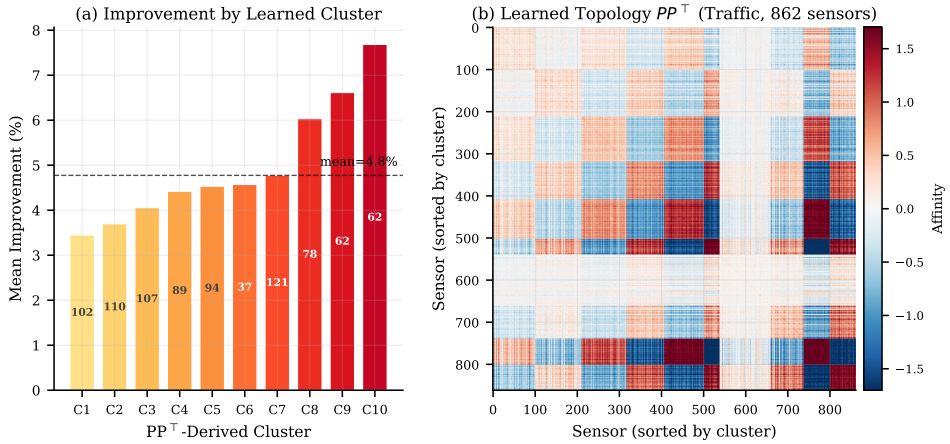

Figure 4: Learned topology on Traffic (862 sensors). (a) Per-sensor improvement by cluster: 2× differential confirms task-relevant structure learned without supervision. (b) $PP^\top$ sorted by cluster shows block-diagonal structure encoding persistent sensor groups.

Electricity (321 clients). At H=192, 306 of 321 clients (95%) improve individually with FTB, with a median per-client improvement of 5.3%. Only 15 clients see any degradation (all mild, <6%). KMeans clustering (k=8) on the learned P matrix reveals clusters with 4× differential improvement: clients in the highest-benefit cluster improve by 16%, while those in the lowest-benefit cluster still improve by 4%. The topology discovers *forecasting-relevant* structure without supervision, grouping variables by their utility for mutual forecasting rather than by statistical similarity. Raw pairwise Pearson correlation between clients does *not* predict which clients benefit most from FTB. Two clients may be highly correlated in raw data but contribute no incremental forecasting value to each other (e.g., both follow the same macro pattern); $PP^\top$ learns to distinguish these cases. Spectral analysis of the learned $PP^\top$ confirms exact rank-6 structure (by construction), with block patterns visible in the first 50 variables that correspond to the discovered clusters.

Traffic (862 sensors). The learned $PP^\top$ sorted by cluster (Figure 4b) exhibits clear block-diagonal structure: dense blocks along the diagonal represent groups of sensors with high persistent mutual affinity, while off-diagonal regions are near zero. The model has discovered groups of sensors with persistent co-variation patterns, plausibly reflecting road network proximity or shared traffic regimes, though we lack sensor metadata to confirm the geographic interpretation.

Comparison to graph methods. This unsupervised structure discovery distinguishes FTB from graph-based methods (Graph WaveNet [14], MTGNN [13]) that require either predefined adjacency matrices or learn them through a separate graph learning module. FTB discovers structure *within* the attention mechanism itself, end-to-end from forecasting loss. The differential improvement across clusters (Figure 4a) confirms the topology is not random noise: variables in high-affinity clusters benefit more from structural priors because they have more persistent relationships to exploit. The 95% improvement rate across variables on both datasets confirms that FTB provides broad, albeit non-uniform, benefit rather than helping a few variables at the expense of others.

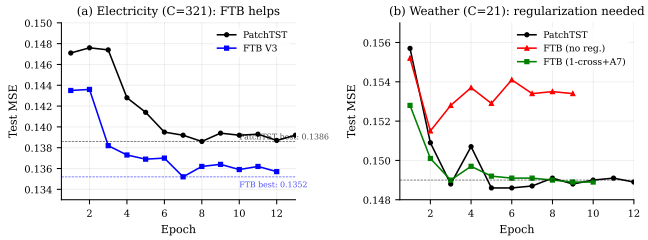

Figure 5: Training dynamics. (a) Electricity (PPP = 3.9): FTB converges below PatchTST by epoch 3. (b) Weather (PPP = 942): unregularized FTB overfits; the constrained variant (green) restores non-degradation.

4.6 Training Dynamics and Gate Behavior

FTB improves optimization beyond final accuracy (Figure 5). On Electricity, FTB achieves lower test MSE than PatchTST from epoch 3 onward, converging in 7 epochs versus 10–15 for the baseline. On Traffic, FTB at epoch 1 already surpasses PatchTST at epoch 3. The Weather panel reveals the complementary failure mode: without regularization, FTB's test MSE worsens after epoch 2 despite decreasing train loss—classic overfitting of the overparameterized cross-attention pathway.

The learned gate g self-calibrates without manual tuning. Starting from $g_0 \approx 0.27$ (conservative initialization), it converges to $g \approx 0.53$ on Electricity, $g \approx 0.47$ on Traffic, and remains at $g \approx 0.28$ on Weather—correctly identifying that Weather lacks cross-variable signal. The per-head topology scaling λ also reveals learned specialization: most heads increase λ from $\sim$2.2 to $\sim$3.0 over training (building topology reliance), while one head maintains $\lambda \approx 1.8$ throughout, acting as a "free" head for non-structural routing. Zeroing $PP^\top$ at

test time after full training degrades MSE by 25–30%, confirming the model permanently co-adapts with the structural prior rather than outgrowing it.

5 Discussion and Conclusion

Why attention-based mixing outscales MLP mixing. TSMixer and FTB share a key architectural insight: interleaved cross-variable mixing improves multivariate forecasting. The difference emerges at scale: TSMixer's channel MLP forces all cross-variable information through a rank-32 bottleneck ($C \to d \to C$), losing fine-grained sub-cluster structure at C=321. FTB's cross-attention computes pair-specific weights without a rank bottleneck, with $PP^\top$ providing a structural scaffold from just 1,926 parameters. Without topology, cross-attention alone provides only -0.9% at H=336 (Table 3)—$PP^\top$ is what makes the pair-specific weights useful rather than noisy.

Practical guidance. Two diagnostics guide FTB deployment, both computable from dataset properties before any training run.

(1) Should I use FTB? Compute $\text{PPP} = \theta_{\text{cross}}/\binom{C}{2}$. With our default architecture (3 cross-attention layers, d_{model}=128), $\theta_{\text{cross}} \approx 198$K. If PPP < 40 ($C > 100$): use FTB, expect 2–5% improvement. If PPP > 100 ($C < 50$): do not use full cross-attention; either use topology-only (just $PP^\top$ bias on existing temporal attention) or skip entirely.

(2) What rank d_p? Compute the number of strong neighbors per variable ($|\{j : |r_{ij}| > 0.5\}|$ for each i), then CoV = std/mean of these counts. If CoV > 0.4 (heterogeneous connectivity, as in Electricity with diverse consumer types), higher d_p toward $\sqrt{C}$ may help at short-medium horizons. If CoV < 0.3 (uniform block structure, as in Traffic with homogeneous sensors), d_p=6 is sufficient and higher rank overfits.

Why not iTransformer? Table 2 shows that at matched capacity (d=128), iTransformer is competitive at H=96 but degrades to PatchTST-level or worse at $H \geq 336$. The mechanism is clear: iTransformer replaces temporal attention with variable attention, sacrificing CI shared-weight regularization. Without a structural prior, its cross-variable attention overfits to transient correlations that do not generalize to longer horizons. FTB preserves CI regularization and adds cross-variable structure through a controlled side-channel; the topology prior $PP^\top$ guides attention toward persistent relationships, which is precisely what enables FTB to maintain gains at long horizons where iTransformer and CrossOnly fail. At higher capacity (d=512), iTransformer outperforms all d=128 methods—but at 4× the parameters and without CI data efficiency. The iTransformer+FTB null result confirms that $PP^\top$ encodes what unconstrained attention discovers given sufficient capacity; FTB's value lies in achieving this with far fewer parameters while preserving regularization.

Scope and applicability. FTB is designed for channel-independent backbones that lack native cross-variable modeling (e.g., PatchTST, DLinear). For architectures that already compute full cross-variable attention (e.g., iTransformer), the topology prior provides no additional benefit, as these models can discover persistent structure on their own given sufficient capacity. In practice, FTB is most effective when $C > 100$ (PPP < 40) and the dataset contains latent group structure among variables.

Conclusion. FTB separates persistent cross-variable structure from dynamic correlations through a factored PSD bias, revealing a precise horizon-dependent complementarity: the topology prior's contribution nearly triples from short to long horizons as temporal signal fades. The Params-Per-Pair diagnostic predicts when structural priors will help, validated by correct positive results on high-C datasets and a correct negative on low-C data. Current results use PatchTST as the backbone; extending to other CI architectures and exploring adaptive topology that evolves over time are natural next steps. All experiments use the public Time-Series-Library [12] codebase; code and hyperparameters (Table 6) will be released upon acceptance. For practitioners with $C > 100$ variables and latent group structure, FTB offers a drop-in improvement with minimal tuning.

Table 6: Hyperparameter summary across datasets.

	Electricity	**Traffic**	**Weather**
Variables C	321	862	21
d_{model} / d_{ff}	128 / 256	128 / 256	128 / 256
Encoder layers	3	3	3
Cross-attn layers	3	3	1 + A7
Topology rank d_p	6	6	6
Learning rate	1e-4	1e-4	1e-4
PPP	3.9	0.53	942

References

[1] Si-An Chen, Chun-Liang Li, Sercan Ö Arık, Nathanael Christian Yoder, and Tomas Pfister. 2023. TSMixer: An All-MLP Architecture for Time Series Forecasting. *Transactions on Machine Learning Research* (2023).

[2] Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. 2017. Implicit Regularization in Matrix Factorization. In *Advances in Neural Information Processing Systems*.

[3] Elon Litman and Gabe Guo. 2026. You Need Better Attention Priors. *arXiv preprint arXiv:2601.15380* (2026).

[4] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*.

[5] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*.

[6] Ofir Press, Noah A Smith, and Mike Lewis. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations*.

[7] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *North American Chapter of the Association for Computational Linguistics*.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Advances in Neural Information Processing Systems*.

[9] Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. 2024. CARD: Channel Aligned Robust Blend Transformer for Time Series Forecasting. In *International Conference on Learning Representations*.

[10] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. 2024. TimeXer: Empowering Transformers for Time Series Forecasting with Exogenous Variables. In *Advances in Neural Information Processing Systems*.

[11] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2024. Unified Training of Universal Time Series Forecasting Transformers. In *International Conference on Machine Learning*.

[12] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

[13] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. 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*.

[14] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *International Joint Conference on Artificial Intelligence*.

[15] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do Transformers Really Perform Badly for Graph Representation?. In *Advances in Neural Information Processing Systems*.

[16] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are Transformers Effective for Time Series Forecasting?. In *AAAI Conference on Artificial Intelligence*.

[17] Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *International Conference on Learning Representations*.

[18] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *International Conference on Machine Learning*.