Attention Mechanisms in Sequence Models for Time-Series Forecasting: A Comparative Study

Rishabh Goenka
EE 344 — Data-driven Modeling and Machine Learning
Departmental Honours Project, Winter 2026

***Abstract*—Attention mechanisms have become the dominant paradigm in neural time-series forecasting, yet most published work benchmarks complete architectures rather than isolating the contribution of the attention mechanism itself. This paper presents a controlled comparative study of three attention variants—Bahdanau additive attention, scaled dot-product self-attention, and Transformer cross-attention—embedded in parameter-matched architectures (∼95K parameters each) and evaluated across three temporally diverse datasets: ETTh1 (multi-scale load cycles), Jena Climate (strong diurnal periodicity), and Exchange Rate (stochastic near-random-walk). At a forecast horizon of 96 steps, no single mechanism dominates. Self-attention achieves the lowest MSE on ETTh1 (0.093 vs. next-best 0.117), the Transformer matches it on Jena while being 6–8× faster at inference, and all neural models fail comprehensively on Exchange Rate, where naive persistence (MSE 0.068) outperforms the best neural model (MSE 0.545). Attention weight analysis reveals near-uniform distributions (normalized entropy ≈ 1.0) across all models and datasets, challenging the common assumption that attention provides interpretable temporal focus. Code is available at https://github.com/rishabhlgoenka/EE344.**

I. Introduction

Time-series forecasting underpins decision-making in energy systems, climate science, and financial markets. Since the introduction of the Transformer architecture [1], attention-based models have rapidly displaced recurrent approaches in many sequence modeling tasks. In forecasting specifically, Informer [5], Autoformer [6], and FEDformer have established attention-augmented architectures as the dominant paradigm on standard benchmarks.

However, most published work evaluates *complete architectures* that differ simultaneously in encoder structure, attention mechanism, decoder strategy, normalization scheme, and parameter count. This makes it difficult to attribute performance differences to the attention mechanism itself rather than to confounding architectural choices. Meanwhile, Zeng et al. [7] demonstrated that a single linear layer can outperform sophisticated Transformer-based forecasters, raising fundamental questions about what attention contributes in the time-series setting.

This paper addresses these questions through a controlled ablation study. Three attention mechanisms—Bahdanau additive attention [2], scaled dot-product self-attention [1], and Transformer cross-attention [1]—are embedded in architectures with matched parameter counts (∼95K) and evaluated under identical training protocols across three datasets deliberately chosen to span the spectrum of temporal characteristics.

The study makes four contributions:

1) Three custom PyTorch architectures implemented from scratch with matched parameter budgets, isolating the attention mechanism as the sole controlled variable.
2) Systematic evaluation across datasets representing periodic, multi-scale, and stochastic temporal regimes.
3) Attention interpretability analysis—entropy, peak-lag, and ACF overlay—testing whether attention learns exploitable temporal structure.
4) Empirical guidelines mapping dataset characteristics to recommended attention mechanisms.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the three datasets. Section IV presents exploratory analysis establishing ground-truth temporal structure. Section V details the model architectures and evaluation methodology. Section VI specifies the experimental protocol. Section VII reports forecasting, efficiency, and interpretability results. Section VIII interprets the findings. Section IX concludes with empirical guidelines and future work.

II. Related Work

A. Transformer-Based Forecasting

The application of attention to time-series forecasting accelerated with Informer [5], which introduced ProbSparse attention to reduce the quadratic cost of self-attention and established the ETTh1 benchmark. Autoformer [6] replaced standard attention with an auto-correlation mechanism operating in the frequency domain. These architectures differ in many dimensions simultaneously—encoder depth, decomposition strategy, normalization, and attention variant—complicating attribution of performance gains to any single component. This paper holds all non-attention components constant to isolate the mechanism's contribution.

B. The Linear-Model Challenge

Zeng et al. [7] showed that a single linear layer (LTSF-Linear) outperforms Informer, Autoformer, and other Transformer-based models on nine benchmarks, arguing that self-attention's permutation invariance causes temporal information loss. Notably, they found that shuffling Transformer inputs on the Exchange Rate dataset produces no performance degradation—consistent with our finding that attention weights on this dataset are near-uniform and that all neural models lose to persistence. Our results both support and nuance

their argument: attention adds no value on random-walk data, but self-attention pre-processing provides genuine benefit on multi-scale periodic data (Section VII).

C. Attention as Explanation

Jain and Wallace [8] demonstrated in NLP that learned attention weights are frequently uncorrelated with gradient-based feature importance and that alternative attention distributions can yield equivalent predictions. Wiegreffe and Pinter [9] challenged this conclusion, arguing that the claim depends on one's definition of explanation. This debate has been conducted almost entirely in natural language processing; our study provides independent evidence from time-series forecasting, where the expected temporal structure (known from autocorrelation analysis) offers a concrete ground truth against which attention patterns can be compared.

D. Attention Mechanism Taxonomy

Bahdanau et al. [2] introduced additive attention for neural machine translation, computing alignment scores via a learned feedforward network over encoder and decoder states. Luong et al. [3] proposed multiplicative (dot-product) alternatives. Vaswani et al. [1] generalized self-attention with multi-head projections and eliminated recurrence entirely. This paper tests one representative from each paradigm: additive decoder attention (M1), scaled dot-product self-attention over encoder states (M2), and full Transformer multi-head self- and cross-attention (M3).

III. Datasets

Three public datasets were selected to span the spectrum of temporal characteristics relevant to the research question. Table I summarizes their properties.

TABLE I: Dataset summary. All features are continuous numerical with zero missing values.

	ETTh1	**Jena Climate**	**Exchange Rate**
Source	Zhou et al. [5]	MPI Biogeochem. / Kaggle	Lai et al. [10]
Records	17,420 (hourly)	420,551 (10-min → 70,041 hourly)	7,588 (daily)
Features	7	14	8
Target	Oil temp. (OT)	Air temp. (°C)	SGD/USD
Character	Multi-scale load cycles	Strong diurnal + annual periodicity	Stochastic, near-random-walk

ETTh1 contains hourly readings from an electricity transformer in China (Jul 2016–Jun 2018), with oil temperature as the prediction target and six power-load features exhibiting daily, weekly, and seasonal dependencies [5].

Jena Climate provides 14 meteorological variables recorded every 10 minutes at a weather station in Jena, Germany (2009–2017). Data is resampled to hourly resolution to align with ETTh1, yielding approximately 70,000 effective samples with strong diurnal and annual cycles.

Exchange Rate contains daily exchange rates for eight currencies against USD (1990–2010) [10]. The data exhibits high stochastic volatility, negligible seasonality, and near-random-walk behavior—properties that make it the hardest forecasting target.

The design rationale is explicit: if attention mechanisms respond differently to periodicity, multi-scale structure, and noise-dominated dynamics, this three-dataset design will reveal it.

IV. Exploratory Data Analysis

EDA was conducted to establish the ground-truth temporal structure against which attention weight patterns are later compared.

A. Raw Time Series

Fig. 1 shows the target variable for each dataset with train/validation/test split boundaries. ETTh1 displays multi-scale load cycling with visible trend shifts. Jena exhibits clean diurnal and annual periodicity. Exchange Rate shows stochastic drift characteristic of a random walk.

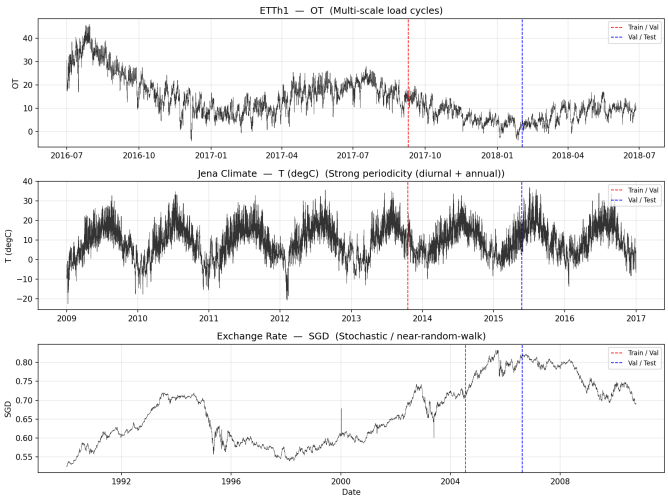

Fig. 1: Raw target variable time series. Dashed vertical lines mark train/validation (red) and validation/test (blue) boundaries.

B. STL Decomposition

Seasonal-trend decomposition using LOESS (STL) [11] was applied to each dataset. Figs. 2–4 show the results. ETTh1 (period=24) reveals clear but complex 24-hour seasonality with visible trend shifts. Jena (period=24, 2015 subset) shows very strong, clean diurnal seasonality with a smooth annual trend. Exchange Rate (period=7) shows negligible weekly seasonality—the residual dominates, confirming the random-walk character.

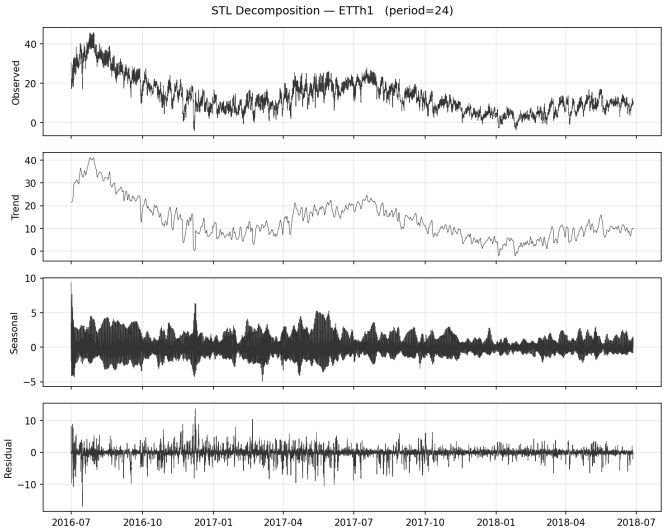

Fig. 2: STL decomposition of ETTh1 (period=24 hours).

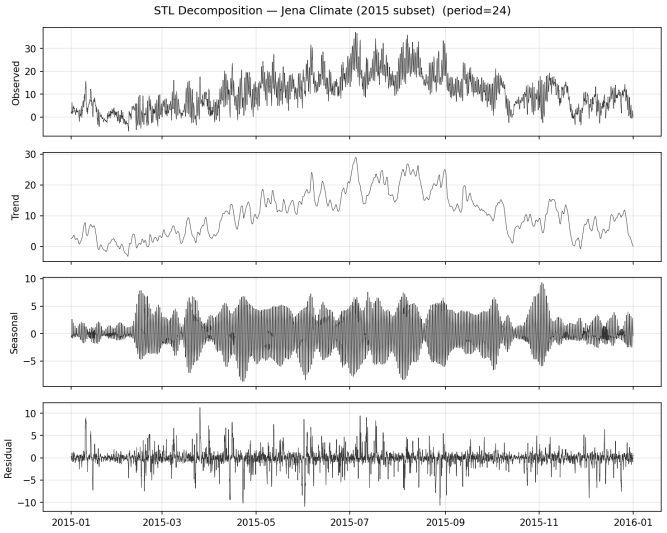

Fig. 3: STL decomposition of Jena Climate, 2015 subset (period=24 hours).

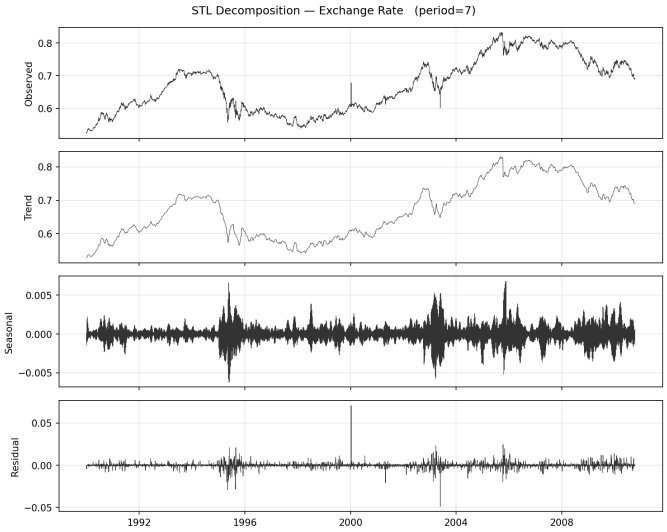

Fig. 4: STL decomposition of Exchange Rate (period=7 days). Negligible seasonality; residuals dominate.

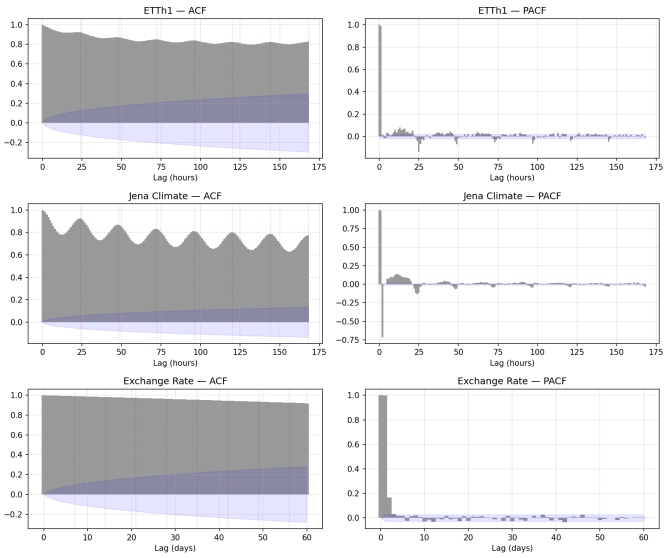

Fig. 5: ACF (left) and PACF (right) for each dataset. Red dotted lines mark seasonal period multiples (24h or 7d).

C. ACF and PACF

Fig. 5 shows autocorrelation and partial autocorrelation functions computed on the training set for each dataset. ETTh1 and Jena exhibit strong ACF peaks at 24-hour multiples, confirming diurnal structure. Exchange Rate's ACF decays slowly and monotonically—characteristic of a unit-root process with no exploitable periodicity.

These ACF profiles establish a testable prediction: if attention mechanisms learn temporal structure, attention peaks should align with ACF peaks on ETTh1 and Jena, and appear diffuse on Exchange Rate. This prediction is evaluated in Section VII-D.

V. Methods

A. Preprocessing

All preprocessing was applied identically across the three model variants per dataset. Key steps include: (i) resampling Jena Climate from 10-min to hourly via mean aggregation; (ii) circular encoding of wind direction in Jena (sin/cos transform, replacing the raw degree column); (iii) chronological train/validation/test splits (60/20/20 for ETTh1 and Jena, 70/10/20 for Exchange Rate); (iv) `StandardScaler` fit on training data only, applied to all splits via `transform()`; (v) sinusoidal cyclical time features (hour-of-day, day-of-week, month-of-year for hourly data; day-of-week, month-of-year for daily); and (vi) sliding window construction with

lookback L=96 for hourly datasets, L=30 for Exchange Rate, and forecast horizon h=96 for all.

The final feature counts after encoding are 13 (ETTh1), 21 (Jena), and 12 (Exchange Rate). Window counts are 10,261/3,293/3,293 (ETTh1), 41,833/13,817/13,818 (Jena), and 5,186/633/1,394 (Exchange Rate) for train/validation/test respectively.

B. Baselines

Three non-neural baselines provide reference points: **Persistence** repeats the last known target value for all 96 forecast steps. **Seasonal Naive** tiles the last seasonal cycle (24h for hourly datasets, 7d for Exchange Rate) across the horizon. **Linear Regression** (scikit-learn [14]) fits a linear map from the flattened lookback window ($L \times d$ features) to 96 output values.

C. Model 1: LSTM + Bahdanau Attention (M1)

M1 is a sequence-to-sequence architecture with a 2-layer LSTM encoder [4] and a Bahdanau additive attention mechanism [2] at each decoder step. The alignment score is $e_{t,i} = \mathbf{v}^\top \tanh(\mathbf{W}_{\text{enc}}\,\mathbf{h}_i^{\text{enc}} + \mathbf{W}_{\text{dec}}\,\mathbf{h}_t^{\text{dec}})$, normalized via softmax to produce attention weights. The decoder is an autoregressive LSTMCell that receives the concatenation of the context vector and the previous prediction at each step, looping 96 times. Teacher forcing is applied with a ratio that anneals linearly from 1.0 to 0.0 over 30 epochs.

D. Model 2: LSTM + Self-Attention (M2)

M2 shares M1's encoder and decoder but inserts a scaled dot-product self-attention layer (single head) over the encoder hidden states *before* decoding: $\text{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\mathbf{Q}\mathbf{K}^\top/\sqrt{d})\,\mathbf{V}$, where $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are linear projections of the encoder output. The decoder then applies Bahdanau attention over the self-attended states, identical to M1. The **only controlled difference** from M1 is this self-attention pre-processing layer, which isolates whether refining encoder representations via self-attention improves forecasting.

E. Model 3: Transformer Encoder-Decoder (M3)

M3 is a Transformer [1] with input projection, fixed sinusoidal positional encoding, a 1-layer `TransformerEncoder` (4-head self-attention + feedforward + LayerNorm), and a 1-layer `TransformerDecoder` (masked self-attention + cross-attention to encoder memory + feedforward + LayerNorm). Crucially, M3 uses 96 learned query embeddings decoded in parallel—no autoregressive loop. A causal mask is applied in decoder self-attention. Learning rate warmup over 5 epochs replaces teacher forcing. This architecture completely eliminates recurrence, making it fundamentally different from M1/M2 at inference time.

F. Parameter Count Matching

Table II shows that parameter counts are matched within ∼2.5K per dataset. M2 uses hidden_dim=60 (vs. 64 for M1/M3) to compensate for the extra self-attention projection matrices. This ensures that observed performance differences are attributable to the attention mechanism, not model capacity.

TABLE II: Trainable parameter counts (∼95K target).

Dataset	M1 Bahdanau	M2 SelfAttn	M3 Transformer
ETTh1	95,361	94,921	94,945
Jena Climate	97,409	96,841	95,457
Exchange	95,105	94,681	94,881

G. Evaluation Metrics

Forecasting accuracy: MSE (primary) and MAE (supporting), reported in StandardScaler-normalized space and inverse-transformed original units. **Training dynamics:** best epoch, total epochs, wall-clock training time, overfit ratio (train MSE / validation MSE at best epoch; ≈1.0 = healthy, ≪1 = memorization). **Efficiency:** per-sample inference latency (ms), measured over 100 forward passes after 10 warmup passes. **Interpretability:** Shannon entropy of decoder attention weights, normalized entropy, peak attention lag index and weight.

VI. Experimental Setup

Table III summarizes the shared and per-model training configuration. All models use Adam [12] with MSE loss, `ReduceLROnPlateau` (patience=4, factor=0.5), gradient clipping (max_norm=1.0), and a maximum of 80 epochs. Random seed is fixed at 42.

TABLE III: Per-model hyperparameters.

Parameter	M1	M2	M3
hidden / d_{model}	64	60	64
Encoder layers	2 (LSTM)	2 (LSTM)	1 (Transf.)
Decoder	LSTMCell	LSTMCell	1 (Transf.)
Heads	—	—	4
d_{ff}	—	—	144
Learning rate	10^{-3}	5×10^{-4}	5×10^{-4}
Weight decay	10^{-4}	10^{-4}	10^{-4}
Dropout	0.2–0.3	0.2	0.2
Teacher forcing	1.0→0.0 / 30 ep	same	None
LR warmup	—	—	5 epochs
Early stop patience	7–10	7–10	7–10
Batch size	32–64	32–64	32–64

Best checkpoint (lowest validation MSE) is restored before test evaluation. All metrics are reported on the held-out test set.

a) Scope limitations.: Experiments use a single forecast horizon (h=96) and a single random seed (42). The original project proposal specified multiple horizons (96, 192, 336, 720) and 3–5 random seeds with significance testing. These were not completed due to computational constraints. Multi-horizon and multi-seed experiments are discussed as future work in Section IX. All reported results should therefore be interpreted as point estimates without error bars.

VII. Results

A. Cross-Dataset Performance

Table IV presents test-set MSE and MAE for all 18 configurations (3 datasets × 6 models) at h=96. Fig. 6 visualizes the

comparison and Fig. 7 shows the neural-model-only heatmap.

TABLE IV: Test-set forecasting performance at h=96 (scaled units). Bold = best per dataset.

Dataset	Model	MSE	MAE
ETTh1	Persistence	0.129	0.274
	Seasonal Naive 24h	0.140	0.288
	Linear Regression	0.117	0.263
	M1 Bahdanau	0.178	0.346
	M2 SelfAttn	**0.093**	**0.235**
	M3 Transformer	0.195	0.366
Jena	Persistence	0.392	0.481
	Seasonal Naive 24h	0.268	0.398
	Linear Regression	7.712	0.498
	M1 Bahdanau	0.187	0.335
	M2 SelfAttn	0.172	0.321
	M3 Transformer	**0.172**	**0.324**
Exchange	**Persistence**	**0.068**	**0.198**
	Seasonal Naive 7d	0.072	0.206
	Linear Regression	0.099	0.240
	M1 Bahdanau	0.545	0.613
	M2 SelfAttn	0.631	0.663
	M3 Transformer	0.683	0.719

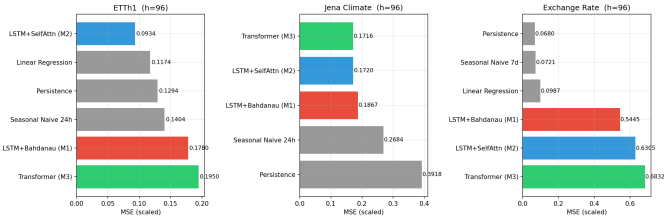

Fig. 6: Test MSE (scaled) by model and dataset at h=96. Neural models excel on ETTh1 and Jena but fail on Exchange Rate.

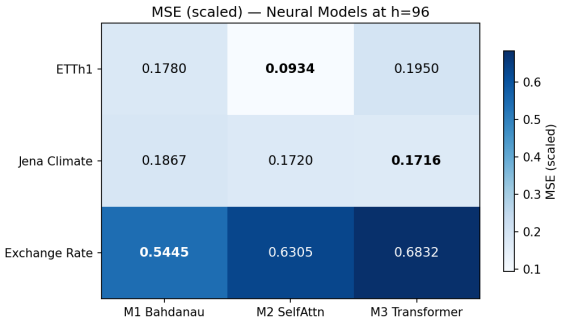

Fig. 7: Neural model MSE heatmap. Bold = best per row. No single mechanism dominates across datasets.

Three patterns emerge. On **ETTh1**, M2 wins decisively (MSE 0.093), beating even Linear Regression (0.117) by 20%—the only dataset where a neural model outperforms all baselines. M1 and M3 both underperform the linear baseline. On **Jena**, M2 and M3 are virtually tied (0.172 vs. 0.172), and all three neural models comfortably beat all baselines. On **Exchange Rate**, all neural models fail catastrophically: persistence (0.068) outperforms the best neural model (M1, 0.545) by an order of magnitude.

a) Linear Regression anomaly on Jena.: LR achieves MSE = 7.71 despite MAE $\approx$ 0.50 (comparable to persistence). This results from fitting unregularized OLS on the flattened lookback window ($96 \times 21 = 2{,}016$ correlated inputs) to predict 96 joint outputs—an ill-conditioned problem where a few large squared errors dominate MSE while median predictions remain reasonable.

B. Training Dynamics

Table V reports convergence statistics. Fig. 8 shows training and validation loss curves.

TABLE V: Training dynamics. Overfit ratio = train MSE / val MSE at best epoch; $\approx$1.0 = healthy, $\ll$1 = memorization.

Dataset	Model	Best Ep.	Train (s)	Val MSE	OF Ratio
ETTh1	M1	1	530	0.125	0.76
	M2	16	1,137	0.079	0.41
	M3	1	167	0.226	1.28
Jena	M1	4	2,032	0.162	0.10
	M2	23	4,996	0.152	0.33
	M3	4	664	0.157	0.99
Exch.	M1	1	191	0.949	0.12
	M2	6	257	1.009	0.01
	M3	4	94	0.980	0.05

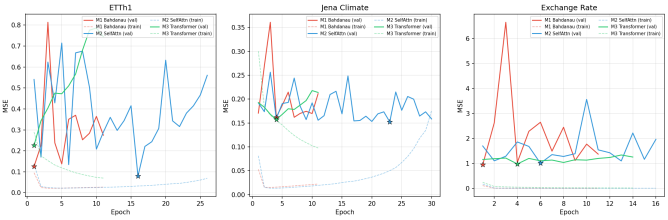

Fig. 8: Training (dashed) and validation (solid) MSE loss curves. Stars mark best-epoch checkpoints.

M1 and M3 achieve their best validation loss at epoch 1 on ETTh1 and Exchange Rate, indicating immediate overfitting (M1) or underfitting (M3, overfit ratio 1.28). M2 consistently needs 2–4$\times$ more epochs (16–23) but produces the lowest validation losses, suggesting that self-attention pre-processing provides a regularizing effect that allows useful learning to continue longer.

M3 trains fastest per epoch due to its parallel decode: 664 s on Jena vs. 4,996 s for M2 (7.5$\times$ speedup), because M1/M2 must execute a 96-step autoregressive loop.

C. Inference Efficiency

Table VI and Fig. 9 report per-sample inference latency.

M3's parallel decode eliminates the 96-step autoregressive loop, yielding 0.12–0.22 ms per sample vs. 0.58–1.53 ms for M1/M2. This 6–8$\times$ speedup is a structural advantage of the Transformer architecture that holds regardless of dataset.

TABLE VI: Per-sample inference latency (ms). M3 is 6–8× faster than M1/M2.

Dataset	Model	Latency (ms)	Std (ms)
ETTh1	M1	1.530	18.58
	M2	1.361	9.29
	M3	**0.179**	0.54
Jena	M1	1.366	5.04
	M2	1.451	15.19
	M3	**0.215**	1.12
Exch.	M1	0.628	1.98
	M2	0.575	1.70
	M3	**0.116**	0.66

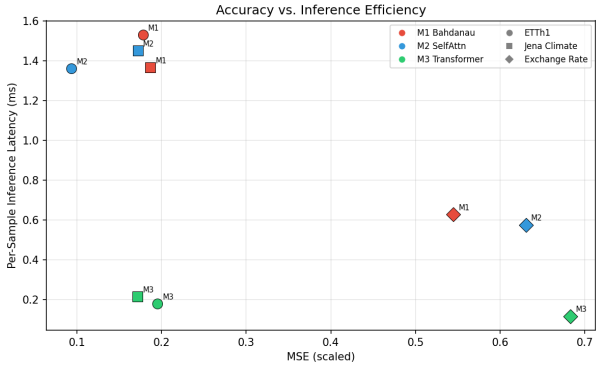

Fig. 9: Accuracy (MSE, x-axis) vs. inference latency (y-axis). M3 (green) consistently occupies the low-latency region. Marker shape indicates dataset; color indicates model.

D. Attention Interpretability

a) Attention weight heatmaps.: Fig. 10 shows attention weight matrices for representative test samples. M1's decoder attention exhibits a slight recency bias on ETTh1 (darker shading toward recent timesteps) and more concentrated patterns on Exchange Rate. M2's self-attention and decoder attention are both visually near-uniform. M3's cross-attention could not be extracted via PyTorch hooks on `nn.MultiheadAttention`—the `need_weights` parameter is not propagated through the `TransformerDecoder` API. This extraction failure is itself a finding about the practical interpretability limitations of standard Transformer implementations.

b) Entropy analysis.: Table VII quantifies the attention weight distributions. Normalized entropy is $\approx$1.0 for all extractable models on all datasets, meaning attention distributes weight nearly uniformly across the lookback window. Peak weights are barely above $1/L$ (uniform baseline: $1/96 \approx 0.0104$ for hourly, $1/30 \approx 0.0333$ for daily). Attention is not learning sharp, sparse patterns.

c) ACF vs. attention overlay.: Fig. 11 presents the central interpretability result: average attention weight distributions overlaid on the autocorrelation function for each dataset. M1's Bahdanau attention (red) shows a monotonically increasing curve on ETTh1 and Jena—a recency bias—rather than tracking the oscillating ACF peaks at 24-hour multiples. M2's self-attention (blue) is flat. Neither mechanism mirrors the known temporal structure.

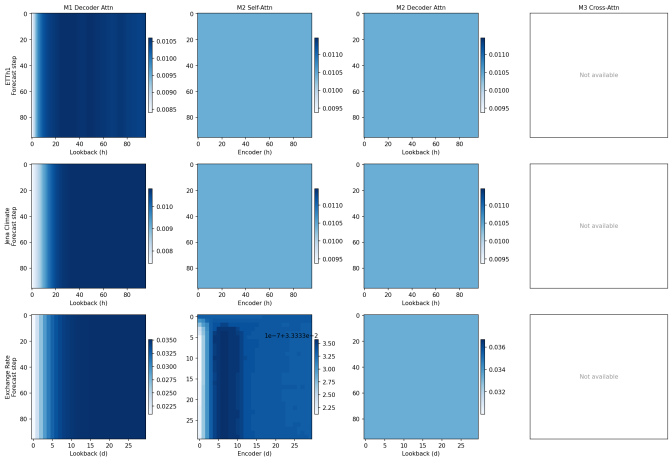

Fig. 10: Attention weight heatmaps. Rows = datasets; columns = M1 decoder, M2 self-attention, M2 decoder, M3 cross-attention (not extractable).

TABLE VII: Attention entropy and peak lag. Normalized entropy $\approx$1.0 indicates near-uniform distributions.

Dataset	Model	Norm. Entropy	Peak Lag	Peak Wt
ETTh1	M1	0.9999	27 h	0.0106
	M2	1.0000	0 h	0.0104
Jena	M1	0.9993	57 h	0.0108
	M2	1.0000	0 h	0.0104
Exch.	M1	0.9982	24 d	0.0351
	M2	1.0000	0 d	0.0333
All	M3	Not extractable		

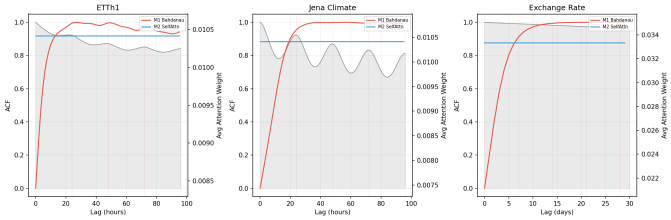

Fig. 11: ACF (gray fill, left axis) vs. average attention weight per lag (colored lines, right axis). Attention does not track ACF peaks.

VIII. Discussion

A. Dataset Character Determines Relative Performance

The central finding is that dataset properties—periodicity, stochasticity, dataset size—predict relative model performance more strongly than the choice of attention mechanism. M2's self-attention pre-processing excels on ETTh1, where complex multi-scale structure benefits from encoder-side refinement but limited data (10K training windows) punishes high-capacity

models that overfit quickly (M1, M3). On Jena, the larger dataset (42K windows) and strong periodicity allow both M2 and M3 to learn effectively, converging to near-identical accuracy. On Exchange Rate, no attention mechanism can extract signal from a near-random-walk—confirming that the value of attention is contingent on the existence of learnable temporal structure.

B. Attention on Random Walks

The Exchange Rate results are consistent with the efficient market hypothesis: if returns are unpredictable, no historical pattern can reduce forecast error below naive persistence. This aligns with Zeng et al.'s [7] finding that shuffling Transformer inputs on Exchange Rate causes no performance degradation—further evidence that attention weights on this dataset are uninformative. The failure of all three neural models (MSE 0.545–0.683 vs. persistence 0.068) is not a deficiency of any specific attention mechanism; it is a fundamental property of the data.

C. Attention as Explanation in Time Series

The near-uniform entropy ($\approx$1.0) across all extractable models and datasets provides time-series evidence supporting Jain and Wallace's [8] NLP finding that attention weights do not provide reliable explanations. Importantly, this does not mean attention is *useless*—M2 outperforms M1, so the self-attention transformation genuinely helps—but it means the attention *weights themselves* are not a reliable window into what the model has learned. The attention-vs-ACF overlay (Fig. 11) makes this concrete: neither M1 nor M2 produces attention peaks at the 24-hour lag multiples that the ACF identifies as the dominant temporal structure.

The failure to extract M3's cross-attention via standard PyTorch APIs highlights a practical barrier to Transformer interpretability that persists even when the architecture theoretically produces attention matrices. Resolving this requires custom forward hooks with `need_weights=True`, which was not achievable within the `TransformerDecoder` wrapper.

D. Why Self-Attention Regularizes

M1 and M3 achieve best_epoch=1 on ETTh1, while M2 continues improving until epoch 16. M2's overfit ratio (0.41) is substantially healthier than M1's (0.76) and M3's underfitting ratio (1.28). A plausible explanation is that the self-attention layer acts as a representation bottleneck: by forcing encoder hidden states to attend to each other before decoding, it produces smoother, more globally-informed representations that prevent the decoder from latching onto sample-specific noise early in training.

E. Limitations

- Single random seed; no error bars or significance tests.
- Single forecast horizon (h=96); degradation at longer horizons not assessed.
- Small model capacity ($\sim$95K parameters); findings may not hold at larger scales or with deeper Transformer stacks.
- M3 cross-attention weights not extractable, limiting the three-way interpretability comparison.
- Benchmark data only; no deployment on live systems.

IX. Conclusion and Future Work

A. Empirical Guidelines

Table VIII synthesizes the findings into actionable recommendations.

TABLE VIII: Empirical guidelines: matching dataset character to attention mechanism.

Characteristic	Model	Reasoning
Strong periodicity + large data	M2 or M3	Both capture periodic structure; M3 benefits from parallel decode
Multi-scale + moderate data	M2	Self-attention pre-processing captures cross-timestep dependencies
Stochastic / random-walk	Persistence	No learnable temporal structure for attention to exploit
Inference speed critical	M3	Parallel decode is 6–8$\times$ faster
Interpretability required	M1 or M2	Native attention weights directly accessible

B. Main Findings

1) No single attention mechanism dominates across temporal regimes.
2) Self-attention (M2) achieves the lowest error on complex multi-scale data (ETTh1 MSE 0.093 vs. next-best 0.117).
3) The Transformer (M3) matches M2 on large periodic data (Jena) while being 6–8$\times$ faster at inference.
4) All neural attention models fail on near-random-walk data, where persistence wins by an order of magnitude.
5) Attention weights are broadly diffuse (entropy $\approx$1.0), not sparsely interpretable—challenging the assumption that attention provides a meaningful window into learned temporal representations.

C. Future Work

Multi-horizon experiments (96, 192, 336, 720 steps) would test whether attention mechanisms degrade differently as the prediction window increases. Multi-seed runs (3–5 seeds) with significance testing would establish reliability of the observed performance differences. Extracting Transformer cross-attention via custom forward hooks would complete the three-way interpretability comparison. Larger model variants and deeper Transformer stacks would test whether the findings scale with capacity. Finally, validation on live industrial or financial data would assess real-world transferability.

References

[1] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 5998–6008, 2017.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, 2015.

[3] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1412–1421, 2015.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] H. Zhou *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. 35th AAAI Conf. Artificial Intelligence*, pp. 11106–11115, 2021.

[6] H. Wu *et al.*, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021.

[7] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proc. 37th AAAI Conf. Artificial Intelligence*, vol. 37, no. 9, pp. 11121–11128, 2023.

[8] S. Jain and B. C. Wallace, "Attention is not explanation," in *Proc. Conf. North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pp. 3543–3556, 2019.

[9] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, pp. 11–20, 2019.

[10] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 95–104, 2018.

[11] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "STL: A seasonal-trend decomposition procedure based on loess," *Journal of Official Statistics*, vol. 6, no. 1, pp. 3–73, 1990.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, 2015.

[13] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 8024–8035, 2019.

[14] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] Max Planck Institute for Biogeochemistry, "Jena Climate dataset," Kaggle, https://www.kaggle.com/datasets/mnassrib/jena-climate.