Attention-Driven Transformers

Pascal Ekin*†

Abstract

Bias-free projection layers combining linear transformations, nonlinear activation, and multiplicative positional encoding can significantly reduce attention requirements when paired with strategically placed attention. This study evaluates such attention-driven architectures on long-horizon time-series forecasting tasks, comparing against PatchTST, N-BEATS, and Temporal Convolutional Networks (TCN) across six standard benchmarks (ETTh1/2, ETTm1/2, Weather, Traffic). Using a configuration with two projection layers per attention layer, the proposed model achieves comparable or improved accuracy while reducing computational cost and parameter count. Projection layers compress and restructure embedding manifolds, while sparse attention restores dimensionality and global organization. Preliminary experiments on autoregressive language modeling suggest that this principle extends beyond forecasting to causal sequence modeling.

1 Introduction

Transformers rely on self-attention to integrate dependencies across sequences. While powerful, this mechanism scales quadratically with sequence length: $O(n^2 d)$ in computation and $O(n^2)$ in memory. This quadratic scaling limits efficiency for long sequences across domains including time-series forecasting and language modeling.

Existing approaches modify attention itself: sparse patterns restrict receptive fields [3, 2], kernel approximations linearize softmax computation

*pfekin@gmail.com
†Independent Researcher

[7, 4], and alternative architectures replace attention with spectral or feed-forward mixing [8, 12, 14]. These methods reduce cost but complicate implementation or sacrifice flexibility.

We do not alter the attention computation itself; rather, we reduce its frequency. The network uses projection blocks (bias-free linear layers with nonlinear activations) for local transformations and places full multi-head attention layers for global coordination. This yields an architecture compatible with standard transformers but markedly lighter. We call this the Attention-Driven Transformer (ADT).

Evaluated on six forecasting benchmarks, ADT achieves state-of-the-art or near–state-of-the-art performance relative to PatchTST, while reducing parameters and computational cost. Results indicate that attention placement can be strategic rather than uniform, without compromising representational capability.

2 Background

2.1 Quadratic Cost of Attention

Given input embeddings $X \in R^{n \times d}$:

$$Q = XW_Q, K = XW_K, V = XW_V,$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V.$$

The similarity matrix $QK^\top \in R^{n \times n}$ incurs $O(n^2 d)$ computation and $O(n^2)$ memory, dominating cost for long sequences.

2.2 Efficiency Strategies

Methods reducing attention cost include sparse patterns that restrict interactions [3, 2], kernel approximations that linearize computation [7, 4], and alternatives that replace attention entirely [8, 12, 14]. ADT retains standard attention but applies it sparingly, using $O(nd)$ projection layers elsewhere.

3 Method

3.1 Projection Layers

Projection layers follow the standard transformer block structure but replace multi-head attention with a bias-free linear projection followed by GELU activation:

$$X' = \text{GELU}(XW), W \in R^{d \times d},$$

$$\tilde{X} = \text{LayerNorm}(X + X'), \hat{X} = \text{LayerNorm}(\tilde{X} + \text{FFN}(\tilde{X})).$$

Residual connections, layer normalization, and the feedforward sublayer follow standard transformer block design. Cost scales as $O(nd)$ with no bias parameters in the projection.

3.2 Attention Layers

Standard multi-head attention with residual connections and feedforward sublayers. Causal masking enforces temporal ordering in autoregressive tasks. Cost is $O(n^2 d)$ per layer, but limiting attention to a small fraction of total layers significantly reduces overall complexity compared to dense-attention models.

3.3 Positional Modulation

$$X_{\text{pos}} = X \odot (P_{\text{learn}} + \beta),$$

where $P_{\text{learn}} \in R^{n \times d}$ contains learned position-specific weights and $\beta \in R$ is a scalar bias. Multiplicative modulation scales input embeddings by position, allowing the model to learn position-dependent feature emphasis.

3.4 Sparse Attention Placement

Attention layers are inserted periodically within the stack. We have tested the following configurations:

- Three-layer: [proj, proj, attn]
- Six-layer: [proj, proj, attn, proj, proj, attn]
- Seven-layer: [proj, proj, proj, attn, proj, proj, proj]

The three-layer configuration is used for primary forecasting experiments (Section 4.1), while the six-layer and seven-layer variant demonstrates scaling to deeper architectures. A six-layer configuration with two attention layers contains 1.38M parameters versus 1.58M for dense PatchTST.

3.5 Forecasting with Patching

Following PatchTST [10], univariate series are segmented into overlapping patches, embedded to d_{model} dimensions, modulated by learned positional encodings, and processed through projection and attention layers before linear projection to the forecast horizon.

3.6 Complexity Analysis

For L total layers and k attention layers:

$$\text{Time} \approx k \cdot O(n^2 d) + (L-k) \cdot O(nd), \text{Space} \approx k \cdot O(n^2) + (L-k) \cdot O(nd).$$

When $k \ll L$, scaling approaches $O(Lnd)$. Configurations with two projection layers per attention layer yield 1.3 to 1.5× speedup on 16-GB GPU.

4 Experiments

4.1 Forecasting

We evaluate on six datasets (ETTh1, ETTh2, ETTm1, ETTm2, Weather, Traffic) against three baselines: PatchTST [10], N-BEATS [11], TCN [1]. The model uses a three-layer [proj, proj, attn] configuration with 128 hidden dimensions and 8 attention heads. Patches are length 16 with stride 8, context window 512, and forecast horizon 96. We train with Adam (learning rate 10^{-4}) and dropout 0.15, averaging over ten runs. All experiments use a Google Colab T4 GPU (16 GB).

Dataset	PatchTST MSE	ADT MSE	Δ (%)	Speedup (×)
Weather	0.1607	0.1548	+3.7	1.45
Traffic	0.3263	0.3206	+1.8	1.38
ETTh1	0.4450	0.4387	+1.4	1.36
ETTh2	0.2438	0.1941	+20.4	1.37
ETTm1	0.3704	0.3295	+11.0	1.34
ETTm2	0.1850	0.1751	+5.4	1.44

Table 1: Forecasting performance across six datasets (mean of ten runs).

ADT matches or exceeds PatchTST on all benchmarks, with particularly strong gains on ETTh2 (+20.4%) and ETTm1 (+11.0%). A six-layer configuration [proj, proj, attn, proj, proj, attn] achieves MSE = 0.1839 on ETTh2 versus 0.2828 for dense attention (35.0% improvement), suggesting sparse attention patterns may scale effectively to deeper architectures, though systematic investigation remains for future work.

Dataset	TCN	N-BEATS	PatchTST	ADT
Weather	0.3679	0.1737	0.1607	**0.1548**
Traffic	0.5141	0.3297	0.3263	**0.3206**
ETTh1	1.5799	0.4642	0.4450	**0.4387**
ETTh2	1.1139	0.2553	0.2438	**0.1941**
ETTm1	0.7694	0.3682	0.3704	**0.3295**
ETTm2	0.7570	0.1807	0.1850	**0.1751**

Table 2: Mean MSE across baseline methods.

4.2 Autoregressive Language Modeling (Exploratory)

Beyond forecasting, we test whether sparse attention with projection-based transformations generalizes beyond forecasting, we draw on prior work on summation-based transformers [6]. That study evaluated small-scale autoregressive models combining projection layers with summation aggregation and sparse attention. Subsequent ablation showed that summation was unnecessary—projection layers with sparse attention alone achieved comparable results—suggesting that ADT is a simpler, more general formulation.

Results from prior work [6]:

Dataset	Dense Attention PPL	Hybrid Summation PPL
WikiText-2	300	**274**
IMDB	150	**145**
AG News	**64**	66
CMU Books	286	**269**

Table 3: Autoregressive performance on small text datasets (from [6]).

Hybrid configurations matched or exceeded dense baselines on three of four datasets, with AG News differences within variance margins.

Implications. These results indicate the attention-driven principle (sparse attention organizing projection-based transformations) applies to both forecasting and causal sequence modeling. However, these experiments used small models (512 hidden dimensions, 512-token context, 4 layers) trained on limited datasets, and validation at large scale remains for future work. These experiments are intended as small-scale proof of principle rather than claims of scalability.

5 Representational Analysis

We analyze layer-wise embeddings using a four-layer language model ([proj, proj, proj, attn], $d_{\text{model}} = 512$, sequence length 512).

We examine two metrics to understand representational dynamics. Cosine similarity between consecutive layers measures how gradually or abruptly representations change through the network. High similarity indicates incremental refinement, while lower similarity suggests more aggressive restructuring. Effective dimensionality, computed via the participation ratio of singular values, reflects how many dimensions actively encode information at each layer. Changes in this measure show whether representations compress or expand.

Dense-attention models show gradual refinement with stable inter-layer similarity and smooth dimensionality changes. Attention-driven models exhibit sharper transitions: lower inter-layer similarity indicates more aggressive restructuring between projection layers, while dimensionality undergoes

a distinct contraction-expansion cycle, compressing at intermediate projection layers before expanding at the final attention layer (Figure 1).

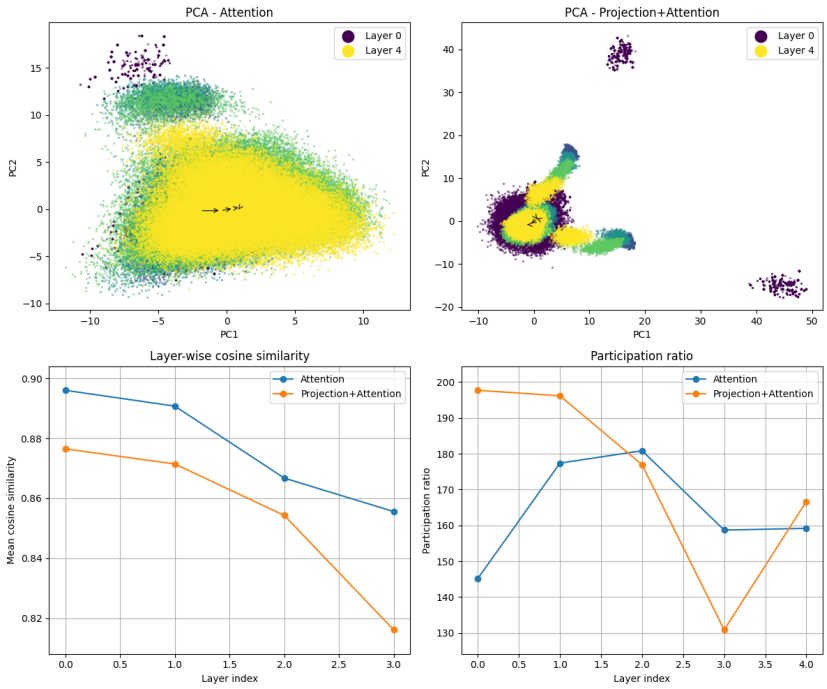

Figure 1: Layer-wise representational dynamics. Top: Principal component trajectories for dense-attention (left) and attention-driven (right) models across layers. Bottom: Cosine similarity between consecutive layers (left) and effective dimensionality measured by participation ratio (right).

This contraction-expansion cycle reveals a division of labor: projection layers compress and restructure representations, while attention expands dimensionality and restores global coherence. This pattern reflects Derrida's *différance* (structure emerging through distinction and deferral) [5]. Projection layers introduce differentiation through compression, and attention defers final resolution, allowing representations to reorganize before stabilizing. This rhythm between local transformation and global coordination distinguishes attention-driven from dense-attention architectures. Viewed abstractly, the alternating structure of projection and attention layers can be interpreted as a differentiable analogue of decision routing—local nonlinear transformation followed by global selection—jointly optimized by gradient

descent.

6 Related Work

6.1 Efficient Attention Mechanisms

Sparse Transformers [3] and Longformer [2] limit receptive fields while retaining quadratic bounds. Linear Transformers [7] and Performers [4] approximate softmax via kernels. FNet [8] and AFT [14] replace attention with deterministic transforms or learned weighting. MLP-Mixer [12] uses feedforward token mixing.

Summation-based transformers. Ekin [6] explored replacing attention with projection layers followed by summation aggregation (cumulative sum in autoregressive settings). Hybrid configurations combining summation with sparse attention matched full-attention performance on small-scale language modeling tasks. ADT removes explicit summation while retaining projection layers and sparse attention, suggesting the simpler architecture may be sufficient at the scales tested.

ADT differs from these approaches by reducing attention frequency rather than altering the mechanism, retaining standard attention but applying it sparingly. Unlike linearized attention models, which approximate the softmax kernel to achieve linear complexity, ADT's efficiency arises from structural sparsity—reducing where attention is applied rather than how it is computed.

6.2 Time-Series Forecasting

PatchTST [10] processes overlapping per-channel patches as tokens, achieving strong benchmark results. iTransformer [9] inverts tokenization by treating variables as tokens and time as embedding dimensions. N-BEATS [11] uses basis expansions for interpretable forecasts. TCNs [1] capture dependencies through dilated convolutions.

ADT preserves PatchTST's patching approach but sparsifies attention, yielding similar or improved accuracy with reduced overhead.

7 Discussion and Future Work

Attention-driven architectures maintain transformer-level representational power with reduced complexity: comparable or better forecasting accuracy with 1.3 to 1.5× speedup and 12% fewer parameters (1.38M vs. 1.58M for six layers). These improvements arise solely from architectural design without altering attention itself.

Future directions include:

1. **Multivariate and multimodal extensions**: integrating cross-variable attention and fusing modalities (text, vision, structured data) into modality-agnostic latent representations.

2. **Deeper networks**: examining sparse-attention scaling beyond 12 layers.

3. **Large-scale autoregressive modeling**: testing whether the principle holds for 7B+ parameter models with 8K+ contexts.

8 Conclusion

Attention-Driven Transformers use projection layers for local nonlinear transformation and attention layers for periodic global coordination. Across six forecasting datasets, this design achieves state-of-the-art or near-state-of-the-art performance while reducing computational cost, memory footprint, and parameter count by 12%. Prior work demonstrated the approach extends to autoregressive language modeling at small scale. These findings suggest that architectural efficiency and representational power need not be mutually exclusive.

Data & Code Availability. All datasets used in this study are publicly available. Implementation and trained model configurations are provided at: `https://github.com/pfekin/attention-driven-transformers`

References

[1] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

[2] Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[3] Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

[4] Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., et al. (2020). Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*.

[5] Derrida, J. (1967). *De la grammatologie*. Les Éditions de Minuit.

[6] Ekin, P. (2025). Summation-based transformers: A path toward linear complexity sequence modeling. *TechRxiv*. `https://doi.org/10.36227/techrxiv.175790522.25734653/v2`

[7] Katharopoulos, A., Vyas, A., Pappas, N., & Fleuret, F. (2020). Transformers are RNNs: Fast autoregressive transformers with linear attention. *arXiv preprint arXiv:2006.16236*.

[8] Lee-Thorp, J., Ainslie, J., Eckstein, I., & Ontañón, S. (2021). FNet: Mixing tokens with Fourier transforms. *arXiv preprint arXiv:2105.03824*.

[9] Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., & Long, M. (2024). iTransformer: Inverted transformers are effective for time series forecasting. *International Conference on Learning Representations (ICLR)*.

[10] Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. *International Conference on Learning Representations (ICLR)*.

[11] Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *International Conference on Learning Representations (ICLR)*.

[12] Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., et al. (2021). MLP-Mixer: An all-MLP architecture for vision. *arXiv preprint arXiv:2105.01601*.

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS 30). arXiv preprint arXiv:1706.03762*.

[14] Zhai, X., Narang, S., Vinyals, O., & Coates, A. (2021). Attention-free transformer. *arXiv preprint arXiv:2105.14103*.