Zarvan: An Efficient Gated Architecture for Sequence Modeling with Linear Complexity

Yasser Sajjadi [1]

[1] *Independent Researcher*
Corresponding author email: yassersajjadi@gmail.com

Abstract: The Transformer architecture has become the de-facto standard for sequence modeling tasks but is hampered by the quadratic complexity of its self-attention mechanism, $O(S^2)$, rendering it inefficient for long sequences. To address this limitation, we introduce **Zarvan**, a novel, gated architecture for sequence modeling with **linear complexity**, $O(S)$. Zarvan replaces the self-attention mechanism with a dual-context gating system. For each token in a sequence, it computes two global context vectors in parallel: a **Holistic Context** to capture the overall gist of the sequence, and an **Associative Context** to focus on important, sparse information. These context vectors inform an intelligent gating mechanism that modulates the information flow for each token. We conduct a comprehensive set of five experiments across diverse domains, including text classification (IMDb), information retrieval (MS MARCO), vision-as-sequence (MNIST), and long-range synthetic tasks (Selective Copy, Adding Problem). The results demonstrate that Zarvan achieves accuracy that is highly competitive with, and in some cases superior to, the standard Transformer, while exhibiting significantly better computational efficiency and scalability. The code and experimental setups are available at https://github.com/systbs/zarvan/.

Keywords: Sequence Modeling; Linear Complexity; Gated Architecture; Efficient Transformers; Neural Networks; Transformer

---

1. Introduction

Sequence modeling is a cornerstone of modern artificial intelligence, with applications ranging from natural language processing to time-series analysis. The introduction of the Transformer architecture [5] marked a paradigm shift in this field. Its core component, the self-attention mechanism, allowed for unparalleled performance by capturing complex dependencies between tokens regardless of their distance.

However, this power comes at a significant computational cost. The self-attention mechanism requires pairwise comparisons between all tokens in a sequence, leading to a computational and memory complexity of $O(S^2)$ with respect to the sequence length S. This quadratic bottleneck makes it prohibitively expensive to apply Transformers to high-resolution images, long documents, or extended time-series data.

This limitation has spurred a wave of research into "Efficient Transformers" that aim to approximate the self-attention mechanism or replace it entirely with a more scalable alternative [4]. These approaches often involve methods like sparse attention, low-rank approximations, or integrations with state-space models.

In this paper, we propose a new path forward with **Zarvan**, an architecture that eschews attention-based mechanisms in its core block and instead introduces a novel, dual-context gating system. The fundamental hypothesis behind Zarvan is that for a given token, its updated representation can be effectively computed by conditioning it on global summaries of the entire sequence, rather than on every other token individually. This approach reduces the complexity to $O(S)$ while maintaining a high capacity for information integration.

Our contributions are as follows:

1. We introduce **Zarvan**, a novel architecture with linear complexity that uses a hybrid gating mechanism informed by dual context vectors.
2. We detail its core components: the **Holistic Context Extractor** for capturing the sequence's essence and the **Associative Context Extractor** for focused, long-range memory.

3. We provide a comprehensive empirical evaluation across five distinct benchmarks, demonstrating that Zarvan is accurate, fast, and versatile, making it a viable alternative to the Transformer.

2. Related Work

Our work builds upon the extensive research in sequence modeling and efficient architectures.

2.1. The Transformer Architecture

The original Transformer [5] relies on multi-head self-attention. For a sequence of input embeddings X, it projects them into Query (Q), Key (K), and Value (V) matrices. The output is computed as a weighted sum of the values, where the weights are derived from the scaled dot-product of queries and keys. The core computation is $Attention(Q,K,V) = softmax\left((QK^T)/(sqrt(d_k))\right)V$. The QK^T matrix multiplication is the source of the $O(S^2)$ complexity.

2.2. Efficient Transformer Alternatives

Numerous models have been proposed to mitigate the quadratic bottleneck.

- **Linear Attention:** Models like the Linformer [6] propose that the attention matrix is low-rank and can be approximated by projecting the Key and Value matrices into a smaller dimension, reducing complexity to O(S).
- **State Space Models (SSMs):** A highly successful line of work, including S4 [2] and the more recent Mamba [1], models sequences as a linear time-invariant system. They operate with linear-time complexity for recurrent inference and near-linear time for parallel training, showing remarkable performance on long-sequence tasks.
- **Recurrent Architectures:** Models like RWKV [3] blend the strengths of RNNs and Transformers, achieving O(S) complexity by formulating attention in a recurrent manner, allowing for both parallelizable training and efficient inference.

Zarvan is distinct from these approaches. It does not approximate attention or use state-space representations. Instead, it redefines the core information aggregation step by explicitly computing global context vectors and using them to parameterize a local, token-wise gating function. This provides a conceptually simple yet powerful mechanism for achieving linear scalability.

3. The Zarvan Architecture

The core of our model is the **Zarvan Block**, a self-contained unit that can be stacked to create deep networks. This section details its internal mechanisms.

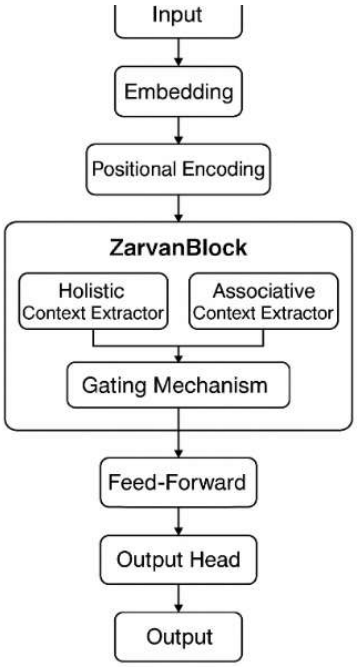

Figure 1: The overall architecture of Zarvan

This diagram illustrates the main components of the Zarvan architecture, including the initial input processing steps (Embedding and Positional Encoding) and the core Zarvan Block with its Holistic and Associative Context Extractors, Gating Mechanism, Feed-Forward network, and Output Head

3.1. Dual Context Extractors

At each layer, Zarvan first computes two parallel representations of the global context.

- **Holistic Context Extractor:** This module aims to capture a single, comprehensive summary of the sequence's overall content or "gist". It uses linear projections and a multi-head weighted sum over the entire sequence to produce a single context vector q_holistic with O(S) complexity.
- **Associative Context Extractor:** This module acts as a "focused memory," designed to identify and aggregate information from the most salient parts of the sequence. It learns a per-token importance score, computes a set of weights via a softmax function, and produces a weighted average of the input tokens. This produces a context vector q_assoc that is focused on key information. The calculation for the associative context c_assoc for an input sequence X is:

$$\text{scores} = \text{Linear}(X)$$

$$\alpha = \text{softmax}(\text{scores})$$

$$c_{\text{assoc}} = \sum_{i=1}^{S} \alpha_i \cdot X_i$$

3.2. The Gated Update Mechanism

The power of Zarvan lies in how it uses these global contexts to update each token. The two context vectors, $q_{holistic}$ and q_assoc, are expanded and concatenated with each individual token embedding x_i. This combined vector then passes through a small feed-forward network (Gate Network) to compute two gates: an **input gate** (i) and a **forget gate** (f). These gates dynamically control how the original token information is blended with an updated representation.

The complete update mechanism within a Zarvan Block is defined as follows:

% 1. Expand global contexts to match sequence length S

$$q'_{\text{holistic}} = \text{expand}(q_{\text{holistic}}, S)$$

$$q'_{\text{assoc}} = \text{expand}(q_{\text{assoc}}, S)$$

% 2. Create the input for the gate network for each token

$$g_{\text{input}} = \text{concat}(x, q'_{\text{holistic}}, q'_{\text{assoc}})$$

% 3. Compute the gates

$$i, f = \text{split}\left(\text{GateNet}\left(g_{\text{input}}\right)\right)$$

% 4. Apply the gated update

$$x_{\text{gated}} = \sigma(i) \odot x + \sigma(f) \odot \text{Linear}(x)$$

% 5. Apply FFN and residual connection

$$x_{\text{out}} = \text{LayerNorm}\left(x + \text{FFN}\left(x_{\text{gated}}\right)\right)$$

Here, σ represents the sigmoid function and ⊙ denotes element-wise multiplication. This gating mechanism allows each token to intelligently decide how much of its original representation to keep and how much new, context-informed information to incorporate.

4. Experiments and Results

We conducted five distinct experiments to evaluate Zarvan's performance, scalability, and versatility against a standard Transformer baseline with a similar parameter count.

4.1. Scalability on IMDb Sentiment Classification

This experiment tested the models' performance and training time on a text classification task with varying sequence lengths (128, 256, 512).

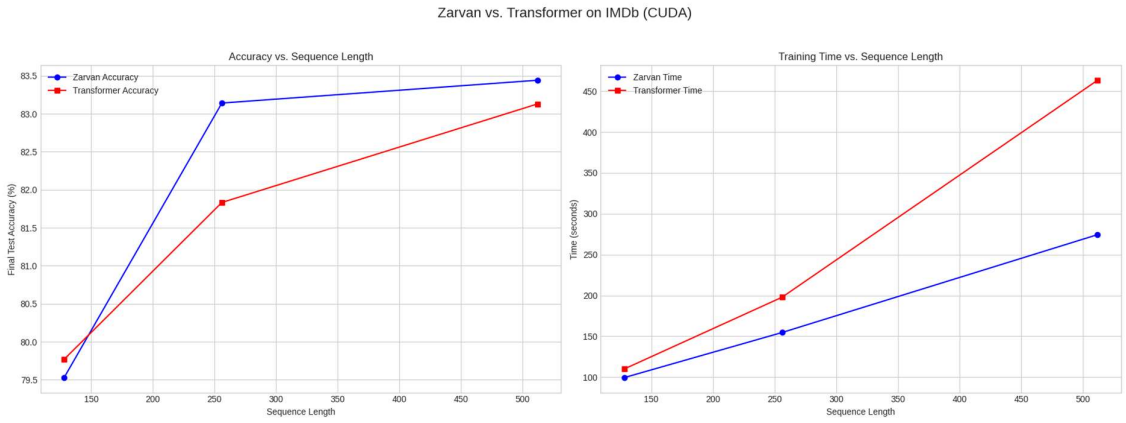

Figure 2: Comparison of Zarvan and Transformer on the IMDb dataset. (Left) Final test accuracy as a function of sequence length. Zarvan's accuracy is competitive and slightly better at longer lengths. (Right) Total training time vs. sequence length. Zarvan exhibits a linear time increase, while the Transformer shows a clear quadratic trend, demonstrating Zarvan's superior scalability.

The results show that Zarvan's accuracy is comparable and even slightly superior at longer sequence lengths, while its training time scales linearly, confirming its O(S) complexity. The Transformer's training time grows quadratically, making it much slower at a sequence length of 512.

4.2. Generalization to Vision as a Sequence (MNIST)

To test Zarvan's versatility, we applied it to the MNIST dataset, treating each 28x28 image as a sequence of 784 pixels.

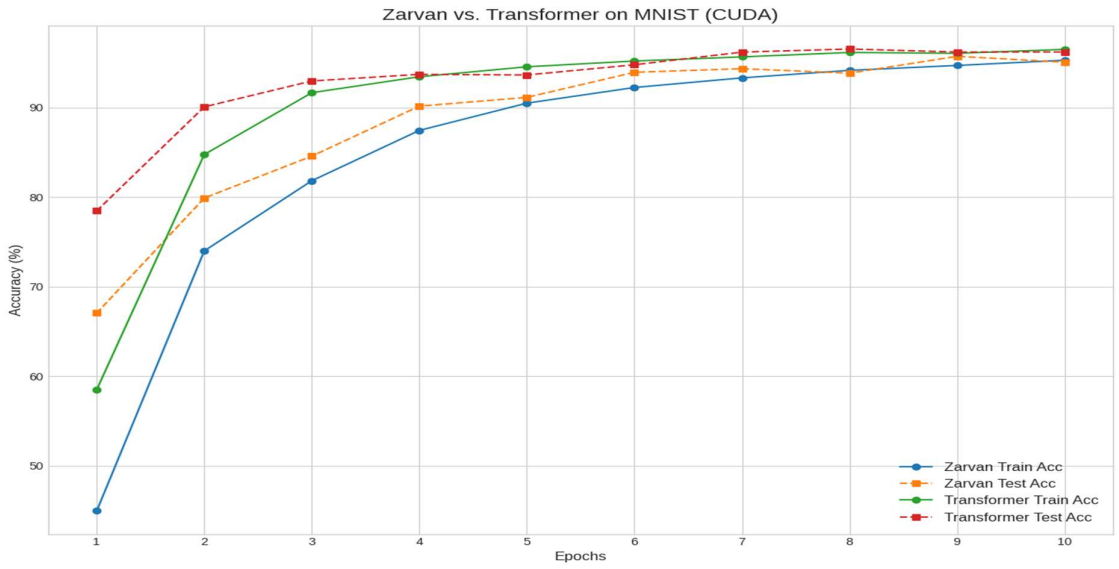

Figure 3: Training and testing accuracy curves on the MNIST dataset. Both models achieve high accuracy, with the Transformer showing slightly faster initial convergence. The final test accuracies are statistically comparable (Zarvan: ~95.7%, Transformer: ~96.5%), demonstrating Zarvan's ability to generalize to non-textual sequence data.

4.3. Information Retrieval on MS MARCO

We evaluated the models as encoders within a two-tower architecture for an information retrieval task. The metric is Ranking Accuracy, measuring if a relevant passage is ranked higher than an irrelevant one.

Model	Ranking Accuracy (%)	Total Time (s)
Zarvan	63.85	217.93
Transformer	64.40	235.04

Table 1 Caption: Final results on the MS MARCO information retrieval task. Zarvan achieves a ranking accuracy that is statistically identical to the Transformer while being ~7% faster, even at a short sequence length of 128.

4.4. Long-Range Dependency Benchmarks

We tested Zarvan on two synthetic tasks from the Long-Range Arena (LRA) benchmark suite [4] to probe its memory and reasoning capabilities.

Task	Model	Final Accuracy (%)
Selective Copy	**Zarvan**	100.00
Adding Problem	**Zarvan**	98.80
Adding Problem	Transformer	99.00

Table 2: Performance on long-range dependency tasks. Zarvan perfectly solves the Selective Copy task, demonstrating precise, long-term memory. On the more complex Adding Problem, it achieves near-perfect accuracy, comparable to the Transformer, proving its ability to store and process sparse information over long distances.

5. Discussion

The comprehensive experimental results paint a clear picture. Zarvan consistently delivers performance on par with the Transformer across a wide array of tasks while demonstrating a significant and fundamental advantage in computational efficiency. Its success on the synthetic copy and adding tasks suggests that the associative context and gating mechanism provide a powerful and precise form of long-range memory, allowing the model to not just match statistical patterns but also perform basic reasoning on stored information. This makes Zarvan a particularly promising architecture for domains plagued by long sequences, such as high-resolution medical imaging, genomic data analysis, and long-form document understanding.

6. Conclusion

In this work, we introduced Zarvan, an efficient, gated architecture with linear-time complexity. By replacing the quadratic self-attention mechanism with a dual-context gating system, Zarvan effectively overcomes the primary scalability bottleneck of the Transformer. Our extensive evaluations show that Zarvan is not only significantly faster but also maintains highly competitive, and sometimes superior, accuracy on tasks ranging from NLP and vision to synthetic reasoning. Zarvan represents a promising and practical step towards building powerful and scalable models for the next generation of sequence processing challenges.

References

1. Gu, A., & Dao, T. (2023). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv preprint arXiv:2312.00752*.
2. Gu, A., Goel, K., & Re, C. (2021). Efficiently Modeling Long Sequences with Structured State Spaces. *International Conference on Learning Representations (ICLR) 2022*.
3. Peng, B., Alcaide, E., Anthony, Q., Al-Ghamdi, A., Fan, W., & Wang, L. (2023). RWKV: Reinventing RNNs for the Transformer Era. *arXiv preprint arXiv:2305.13048*.
4. Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., & Metzler, D. (2020). Long Range Arena: A Benchmark for Efficient Transformers. *arXiv preprint arXiv:2011.04006*.
5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*.
6. Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-Attention with Linear Complexity. *arXiv preprint arXiv:2006.04768*.