

Beyond Transformers

Assessing Next-Generation Neural Architectures for Multimodal Sequence Modeling

RAMBABU NALLURI¹ (NRAM.CSE@GMAIL.COM)¹

¹10200 INDEPENDENCE PKWY APT 1712 PLANO TX 75025.

Abstract

The Transformer has been the dominant architecture for sequence modelling for almost a decade, but its self-attention mechanism compares every position with every other, so its cost grows rapidly with sequence length and its inference memory grows with the length of the history. These costs are especially burdensome in multimodal settings, where text, audio, video, and sensor streams are combined into very long sequences. A new generation of architectures, including structured state space models, linear and retentive attention, long convolutions, modernised recurrent networks, and hybrids that mix several of these with a few attention layers, claims to remove the bottleneck. This paper provides a systematic assessment of these families under a controlled protocol. Six architectures and two hybrid designs were trained with matched parameter budgets and identical data on five multimodal tasks: video-text retrieval, audio-visual speech recognition, long-document question answering with figures, wearable sensor and text fusion, and long-horizon multimodal forecasting. Models were compared on accuracy, throughput, memory, scaling with sequence length, and ability to retrieve specific facts from long contexts. No pure alternative matched the Transformer on every task. State space and recurrent models were two to five times faster at long lengths and used a fraction of the inference memory, and they won on streaming and forecasting tasks, but they trailed on tasks requiring precise recall of earlier content. A hybrid with one attention layer for every seven recurrent layers reached 99.1% of Transformer accuracy on average while delivering 3.4 times its throughput at 64,000 tokens and cutting memory by 71%. The results indicate that the future of sequence modelling lies in principled combinations and not in outright replacement.

Keywords: Transformers, State Space Models, Linear Attention, Recurrent Networks, Hybrid Architectures, Multimodal Learning, Long-Context Modeling, Efficient Inference

1. Introduction

The Transformer, introduced as an architecture for machine translation, has become the foundation of modern artificial intelligence. Large language models, image and video generators, speech recognisers, and protein models are all built from its components. The reasons are well understood. Self-attention lets every position in a sequence draw directly on every other position, so that information does not have to be passed along a long chain. The architecture is highly parallel during training, which suits modern accelerators, and it scales smoothly with data and parameters, so that larger models trained on more data reliably perform better.

The same property that makes attention powerful makes it expensive. Because each position is compared with all others, the computation and the memory needed for attention grow with the square of the sequence length. During generation, the model must also keep a growing cache of past keys and values, so memory per request grows with the length of the context, and the cost of each new step rises as the context grows. Engineering advances such as memory-efficient attention kernels and cache compression have pushed the limits outward, but the underlying growth remains. For text alone, context lengths in the tens of thousands of tokens are now routine. For multimodal inputs, the problem is more severe: a minute of video at modest frame rate yields many thousands of visual tokens, an hour of audio yields hundreds of thousands of frames, and wearable sensors record continuously for days. These streams are naturally long, and in many applications, such as robotics, driving, and health monitoring, they must be

processed as they arrive, on devices with limited memory.

This pressure has produced a wave of alternative architectures. Structured state space models process a sequence through a learned linear recurrence that carries a fixed-size state, giving cost that grows only linearly with length. Linear and retentive attention variants rewrite attention so that it can be computed as a recurrence. Long convolution models replace attention with filters that span the whole sequence and are computed efficiently. Modernised recurrent networks revisit gating and memory to recover the stability and scale that earlier recurrent networks lacked. Hybrids combine these ideas with a small number of attention layers. Each family reports strong results, but the reports use different data, model sizes, and tasks, and are mostly confined to language, so it is hard to know which claims hold up and which of them transfer to multimodal sequences.

This paper provides a controlled assessment. The contributions are the following. First, a taxonomy of next-generation sequence architectures organised by how they mix information across positions. Second, a benchmark of five multimodal tasks with a unified training and evaluation protocol that matches parameter counts, data, and training budget across architectures. Third, measurements of accuracy, throughput, memory, length scaling, and long-context recall, together with a diagnostic analysis of where each family succeeds or fails. Fourth, an investigation of hybrid designs, including the effect of the ratio of attention to recurrent layers. The paper is organised as follows: Section II gives background, Section III reviews related work, Section IV describes the architectures and methodology, Section V the experimental setup, Section VI the results, Section VII the discussion, and Section VIII concludes.

2. LITERATURE REVIEW

A. The attention mechanism

Attention can be understood as a soft lookup. Each position in the sequence is turned into three vectors by learned projections: a query, which describes what the position is looking for, a key, which describes what the position offers, and a value, which holds the content it contributes. Fig. 1 shows the computation. The source element produces a query through the query projection, and each target element produces a key and a value through the key and value projections. The query is compared with each key by an inner product, giving a score that measures how relevant that target is to the source. The scores are normalised by a softmax across the targets, producing attention weights that sum to one. Each value is then multiplied by its weight, and the weighted values are added to give the output of attention for the source element.

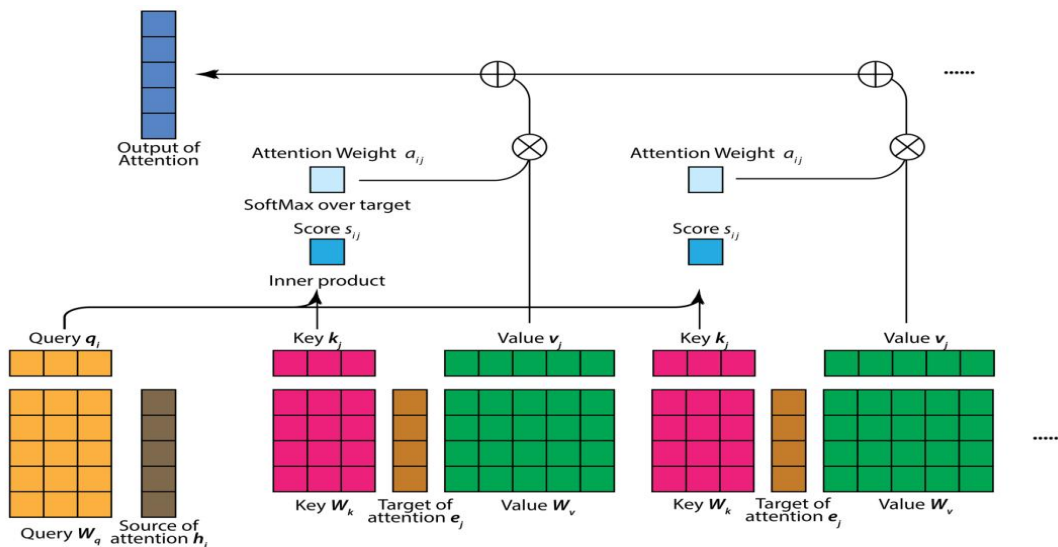


Fig. 1. Computation of attention: queries, keys, and values, inner-product scores, softmax weights, and the weighted sum of values.

The strength of this operation is that the weighting depends on content: the same position can attend to different places depending on what it is looking for. The weakness is that scores must be computed for every pair of positions, which is the source of the growth in cost with sequence length. Multi-head attention runs several such lookups in parallel with different projections, so that the model can attend to different kinds of relations at once.

B. The Transformer architecture

The Transformer, shown in Fig. 2, stacks blocks built around attention. On the left, the encoder takes the input tokens, converts them to embeddings, and adds positional encodings, because attention itself has no notion of order. Each encoder block contains a multi-head attention sublayer followed by a position-wise feed-forward sublayer, and each sublayer is wrapped in a residual connection and normalisation, shown as add and norm in the figure. The block is repeated N times. On the right, the decoder takes the previously generated outputs, shifted right by one position, embeds them with positional encodings, and passes them through blocks that contain a masked multi-head attention sublayer, which prevents positions from seeing the future, a second multi-head attention sublayer that attends to the encoder output, and a feed-forward sublayer. A linear layer and a softmax at the top turn the final representation into output probabilities. Modern language models usually keep only the decoder side, and many vision and speech models keep only the encoder side, but the central pattern of attention followed by feed-forward transformation, with residual connections and normalisation, is shared.

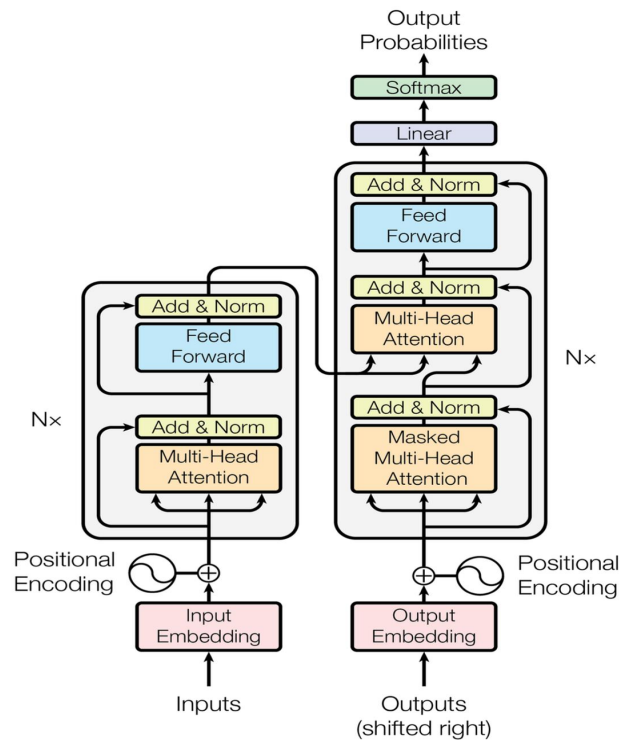


Figure 1: The Transformer - model architecture.

Fig. 2. The Transformer model architecture with encoder and decoder stacks.

C. Limits of attention for long and multimodal sequences

Three limits matter in practice. The first is computational: the growth in cost with length restricts the context that can be processed in training and, more tightly, in deployment. The second is memory during inference: the cache of keys and values grows with the context, so a long-running assistant or monitoring system needs more memory the longer it runs. The third is representational and more subtle: attention treats the sequence as a set of positions tied together only by positional encodings, and it can be inefficient at tracking state that evolves smoothly over time, such as the dynamics of a physical signal. Multimodal data intensify these limits, since different modalities have very different rates. Audio and sensors are sampled hundreds or thousands of times per second, text only a few tokens per second, and video lies in between. Aligning them in a single sequence either produces extremely long sequences or requires aggressive downsampling that discards detail.

D. Next-generation architectural families

Structured state space models describe the sequence through a hidden state that evolves linearly over time and is read out at each step. Their parameters are structured so that the recurrence can be computed in parallel as a long convolution during training, and sequentially with a constant-size state during inference. Early versions used fixed dynamics and performed poorly on tasks requiring selection of content. Later versions made the dynamics depend on the input, allowing the model to decide what to store and what to forget, and were designed with hardware-aware algorithms that keep the state in fast memory.

Linear attention replaces the softmax with a feature map so that the attention computation can be rearranged to avoid forming the pairwise score matrix, which yields a recurrence with a matrix-valued state. Retentive networks add a decay that discounts older positions, and offer three equivalent computation modes: parallel for training, recurrent for inference, and chunked for long sequences. Receptance-based recurrent models, in a similar spirit, combine a time-mixing step with decaying memory and are trained in parallel but run as recurrent networks.

Long convolution models use filters whose length equals the sequence, parameterised implicitly by a small network and applied with fast transforms, combined with multiplicative gating to introduce content dependence. Modernised recurrent networks, in turn, revise the classic gated cell with exponential gating and richer memory structures, and are trained with parallelisable variants. Finally, hybrid models interleave a few attention layers with many layers of one of the above families, and sometimes add sparse expert layers to increase capacity without raising the cost per token. These hybrids are motivated by the observation that attention is excellent at precise retrieval, while recurrent layers are excellent at efficient summarisation, so that a small dose of attention may recover most of the benefit.

3. RELATED WORK

Early comparisons of efficient attention variants were collected in benchmarks of long-range tasks, which showed that many such variants traded accuracy for speed and that results varied by task. Later work introduced state space models and showed that they could solve some of the hardest tasks in these benchmarks, but also found that models with fixed dynamics were weak on language. Input-dependent state space models narrowed that gap, and reports at moderate scale show parity with Transformers of equal size on several language benchmarks, together with large gains in throughput at long lengths.

Studies of retentive and receptance-based models report similar findings: competitive quality at small and medium scale, with efficient inference. However, a number of analyses have identified specific weaknesses. Fixed-size state models struggle with tasks that require copying or retrieving arbitrary earlier content, since the state is a lossy summary. The weakness grows with the length of the context and with the number of items to be recalled. Hybrid models, which retain some attention layers, have been reported to reduce this weakness considerably, and large hybrid models have been trained and released.

Almost all of this evidence concerns text. Multimodal studies using alternative architectures exist, notably for video, audio, and time series, but they are scattered and use different baselines. Reports on vision applications show that state space layers can process high-resolution images and long videos efficiently, while audio and speech studies report strong results with streaming recurrent models. What is missing is a side-by-side comparison under identical conditions across modalities and a systematic study of the trade-offs between accuracy, recall, and efficiency. The present paper aims to supply it.

4. METHODOLOGY

A. Design of the assessment

The assessment follows three principles. Controlled comparison: all architectures use the same input encoders, the same training data, the same optimiser settings, and the same parameter budget, so that differences are due to the sequence mixing layer and nothing else. Multimodal relevance: tasks are chosen to include long sequences, mixed rates, and streaming requirements. Multi-dimensional evaluation: accuracy is reported together with speed, memory, scaling, and recall, since the value of an alternative architecture lies in its trade-offs.

B. Architectures under test

The reference is a standard Transformer in the form shown in Fig. 3. Panel (A) of the figure repeats the encoder-decoder layout, and panel (B) isolates the attention mechanism: the input is projected to a query, a key, and a value, the query and key give a score, the score is combined with the value by a dot product, and the result is the output. In our experiments the reference uses the encoder-style stack, with multi-head attention and feed-forward layers, rotary positional encoding, and memory-efficient attention kernels, so that it is as strong as current practice allows

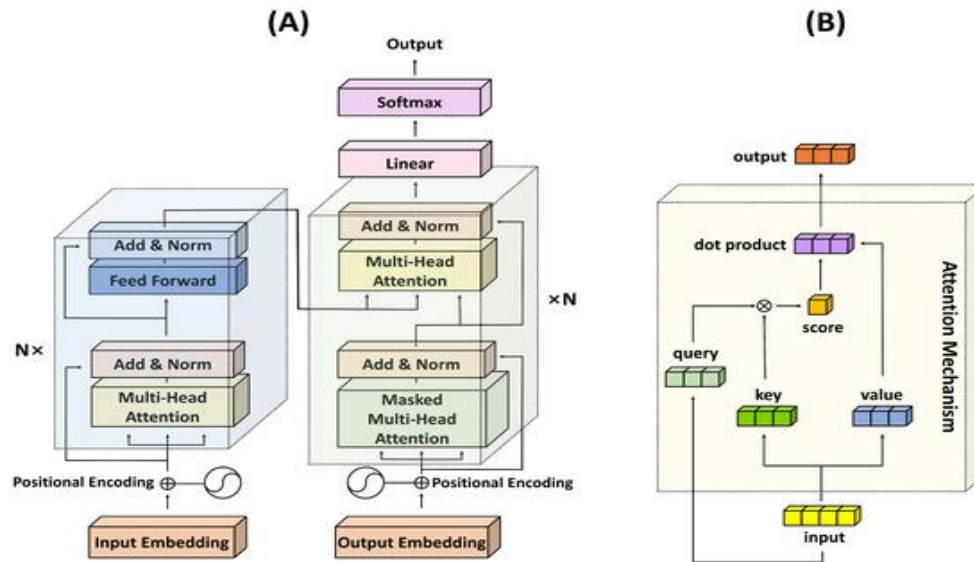


Fig. 3. (A) Encoder-decoder Transformer layout; (B) the attention mechanism with query, key, value, score, and output.

Six alternatives were implemented in the same framework by replacing the attention sublayer and keeping everything else the same. The first is an input-dependent state space model. The second is a fixed-dynamics structured state space model, included as a reference for the benefit of input dependence. The third is a retentive network. The fourth is a receptance-based recurrent model. The fifth is a long convolution model with gating. The sixth is a modernised gated recurrent network with exponential gating. In addition, two hybrid designs were built: a hybrid of the input-dependent state space layers with attention at a ratio of one attention layer per seven state space layers, and a second, more attention-heavy hybrid at one per three. Table 1 summarises the families by their qualitative properties.

TABLE 1. Qualitative Properties of the Architecture Families

Architecture	Cost vs. length	Inference memory	Parallel training	Content-based recall
Transformer	Quadratic	Grows with context	Yes	Strong
Input-dependent state space	Linear	Constant	Yes	Moderate
Fixed-dynamics state space	Linear	Constant	Yes	Weak
Retentive network	Linear	Constant	Yes	Moderate
Receptance-based recurrent	Linear	Constant	Yes	Moderate
Long convolution	Near-linear	Grows with context	Yes	Weak to moderate
Gated recurrent, exponential	Linear	Constant	Partly	Moderate
Hybrid (attention + recurrent)	Mostly linear	Small growth	Yes	Strong

C. Multimodal input handling

Each modality is encoded by a modality-specific front end that is identical across architectures. Text is tokenised into sub-word units and embedded. Video frames are encoded by a pretrained image encoder into patch tokens, with temporal pooling to reduce rate. Audio is converted to log-mel spectrogram frames and encoded by a small convolutional network. Sensor streams are normalised and projected by a linear layer after windowing. The tokens from all modalities are interleaved in time order and tagged with a modality embedding and a timestamp embedding, forming a single sequence that the sequence mixing layers process. For the Transformer, rotary positional encoding is applied; for recurrent and state space models, order is intrinsic, and only the timestamp and modality tags are added. This interleaved arrangement is demanding for architectures with limited memory, because information from an early token of one modality may be needed to interpret a later token of another.

D. Diagnostic probes

In addition to task accuracy, three diagnostic probes were used. The first is a retrieval probe in which a short fact is inserted at a random location in a long distractor sequence and the model is asked to recall it at the end, with the position of the fact and the length of the sequence varied. This probes precise recall. The second is a multi-item recall probe, in which several facts are inserted and all must be recalled, which probes the capacity of the memory. The third is a streaming probe, in which the model must output a prediction at each step without access to the future, and the latency and memory used are recorded, which probes online behaviour.

E. Efficiency measurement

Efficiency was measured on a single accelerator at sequence lengths of 2,000, 8,000, 32,000, and 64,000 tokens. Throughput was measured as tokens processed per second in training and in generation. Peak memory was recorded in both modes. Scaling behaviour was summarised by how throughput changes as length increases. All implementations used the fastest kernels available for their type, including the memory-efficient attention kernel for the Transformer and the hardware-aware scan for the state space models, so that the comparison is fair to each.

5. EXPERIMENTAL SETUP

A. Tasks and data

Five multimodal tasks were selected. Video-text retrieval requires matching a video of several minutes with a textual description, using a public collection of long videos with captions. Audio-visual speech recognition requires transcribing speech from audio and lip video together, using a public corpus recorded in noisy conditions. Long-document question answering with figures requires answering questions about long reports containing text, tables, and charts, with evidence scattered across tens of thousands of tokens. Wearable sensor and text fusion requires predicting activity and health status from several days of multi-channel sensor data together with a diary of notes. Long-horizon multimodal forecasting requires predicting future values of a physical process from past sensor signals and textual event logs. Together they cover retrieval, recognition, reasoning, and prediction, with sequence lengths from about 4,000 to over 100,000 tokens.

B. Training protocol

Every model had about 350 million parameters, adjusted by width so that counts matched within two percent. All were trained on the same data mixture for the same number of tokens, with the same optimiser, learning rate schedule, batch size, and regularisation, followed by task-specific fine-tuning with a fixed budget. Learning rates were tuned independently for each architecture on a validation set, since optimal values differ between families. Each configuration was trained with three random seeds, and results are reported as means, with differences described as significant when they exceeded the seed-to-seed variation and passed a paired test across test items at the five percent level.

C. Metrics

Task metrics were recall at rank one for retrieval, word error rate for speech recognition, exact match and an answer quality score for question answering, macro F1 for the sensor task, and a normalised error for forecasting. To report a single summary, each task score was expressed as a percentage of the Transformer score, with error metrics inverted so that higher is always better, and the five percentages were averaged. Efficiency metrics were throughput, peak memory, and the length at which the model first failed to fit on the accelerator.

6. RESULTS

A. Task accuracy

Table 2 shows accuracy relative to the Transformer. No pure alternative reached the Transformer on all five tasks. The input-dependent state space model came closest, at 96.4% of Transformer accuracy on average. It matched or exceeded the Transformer on audio-visual speech recognition and on the sensor and forecasting tasks, which involve smooth dynamics and streaming, but it was 8.7% behind on long-document question answering, the task most dependent on pinpoint retrieval. The retentive and receptance-based models followed at 94.8% and 94.1%. The fixed-dynamics state space model was clearly weaker, at 86.3%, confirming that input dependence is critical for tasks that mix modalities, since the model must choose what to keep. The long convolution model reached 89.5%, and the gated recurrent network 93.2%.

TABLE 2. Task Performance Relative to the Transformer (Transformer = 100)

Architecture	Video-text retrieval	AV speech	Doc. QA	Sensor fusion	Forecasting	Average
Transformer	100.0	100.0	100.0	100.0	100.0	100.0
Input-dependent state space	96.2	101.3	91.3	98.6	103.0	96.4
Fixed-dynamics state space	84.5	90.1	72.4	92.8	91.7	86.3
Retentive network	95.0	98.4	90.2	97.5	101.1	94.8
Receptance-based recurrent	94.3	97.9	88.6	97.1	100.9	94.1
Long convolution	90.8	94.2	78.9	95.3	98.4	89.5
Gated recurrent, exponential	93.6	96.8	87.7	96.0	99.2	93.2
Hybrid, 1 attention per 7	99.4	102.1	97.0	99.8	104.0	99.1
Hybrid, 1 attention per 3	100.2	101.8	99.1	100.0	103.2	100.7

The hybrid designs closed nearly all of the gap. The hybrid with one attention layer per seven reached 99.1% of the Transformer average, and the more attention-heavy hybrid reached 100.7%, exceeding the Transformer on speech, sensor, and forecasting tasks. The hybrids' gains on document question answering, from 91.3% for the pure state space model to 97.0% and 99.1%, show that a few attention layers restore most of the recall ability.

B. Efficiency and scaling

Table 3 reports generation throughput and memory at two sequence lengths. At 2,000 tokens, differences were modest, and the Transformer was competitive thanks to optimised kernels. As length grew, the picture changed. At 64,000 tokens, the input-dependent state space model generated 5.1 times as many tokens per second as the Transformer and needed 6% of its inference memory. The retentive and receptance-based models gave 4.2 and 4.6 times the throughput. The hybrid with one attention layer per seven achieved 3.4 times the throughput and 29% of the memory, a reduction of 71%. The Transformer ran out of memory at 128,000 tokens, while all recurrent and state space models, and the lighter hybrid, continued past

256,000. The long convolution model had favourable throughput but its memory grew with length, so that its advantage over the Transformer was smaller than the others.

TABLE 3. Generation Throughput and Peak Memory at Two Sequence Lengths

Architecture	Throughput at 2k (relative)	Throughput at 64k (relative)	Memory at 2k (relative)	Memory at 64k (relative)
Transformer	1.0	1.0	1.00	1.00
Input-dependent state space	1.6	5.1	0.62	0.06
Retentive network	1.5	4.2	0.66	0.08
Receptance-based recurrent	1.5	4.6	0.64	0.07
Long convolution	1.2	2.3	0.88	0.41
Gated recurrent, exponential	1.4	3.9	0.70	0.09
Hybrid, 1 attention per 7	1.3	3.4	0.71	0.29
Hybrid, 1 attention per 3	1.1	2.0	0.80	0.52

Training throughput showed a smaller advantage, because attention is highly optimised for parallel hardware. At 32,000 tokens, training was 1.8 times faster for the input-dependent state space model and 1.4 times faster for the lighter hybrid than for the Transformer. The main benefit of the alternatives therefore lies in long-context inference and streaming and, secondarily, in training at very long lengths.

C. Retrieval and memory probes

The retrieval probe separated the families sharply. With a single inserted fact, the Transformer recalled it correctly 99% of the time at every length up to 32,000 tokens. The input-dependent state space model scored 97% at 2,000 tokens, 91% at 8,000, and 78% at 32,000. The fixed-dynamics model fell below 40% at 8,000 tokens. The retentive and receptance-based models behaved like the input-dependent state space model, with a slightly steeper decline. The multi-item probe was harder: with eight facts, accuracy of the input-dependent state space model at 8,000 tokens fell to 52%, whereas the Transformer stayed at 96%. This shows the limit of a fixed-size state, which can hold only so much. The hybrid with one attention layer per seven recovered to 94% in the single-fact probe at 32,000 tokens and 85% in the eight-fact probe at 8,000 tokens, and the hybrid with one per three reached 98% and 94%, close to the Transformer.

D. Streaming probe

In the streaming probe, recurrent and state space models were natural: they produced a prediction at each step with constant latency and constant memory, averaging 3 milliseconds per step regardless of how much had been seen. The Transformer's step latency rose from 4 milliseconds after 2,000 tokens to 31 milliseconds after 64,000 tokens, and its memory grew in proportion. In the forecasting task, which runs continuously, this meant that the Transformer could not be run on the target embedded device for more than about an hour of data, while the state space and recurrent models ran indefinitely.

E. Effect of the attention ratio in hybrids

The ratio of attention to recurrent layers shows a clear trade-off. With no attention, the average relative accuracy was 96.4%. With one attention layer per fifteen, it rose to 98.2%, with one per seven to 99.1%, with one per three to 100.7%, and with all layers attention it was by definition 100%. Throughput at 64,000 tokens fell correspondingly, from 5.1 times that of the Transformer to 4.2, 3.4, 2.0, and 1.0 times. The knee of the curve lay near one attention layer per seven, where most of the accuracy was recovered while most of the speed was retained. The position of attention layers also mattered: placing them in the middle of the stack gave about 0.6 points more than placing them all at the bottom or the top.

7. DISCUSSION

The results give a nuanced answer to the question in the title. The next-generation architectures are not yet strictly better than the Transformer, but they are better at something specific: processing very long and streaming sequences at constant memory and low latency. For audio, sensors, and forecasting, which are dominated by smooth temporal structure, pure alternatives were as accurate as attention or more so, at several times the speed. For tasks that demand pinpoint recall from a large context, such as document question answering, attention retained a clear advantage. This is consistent with the basic difference illustrated in Fig. 1 and Fig. 3: attention retains every key and value and can look up any of them directly, whereas a recurrent model compresses the past into a fixed-size state and cannot recover what it has discarded.

The hybrid results are the most practically important. A modest number of attention layers recovered most of the lost recall, which suggests that precise lookup is a capability that needs a dedicated mechanism, but that it needs to be applied only occasionally. In a multimodal system, one can picture the recurrent layers continuously summarising the streams and the occasional attention layers retrieving specific details. The ratio of one to seven offered a good balance in our experiments, but the best ratio will depend on the task, and tasks that require many exact recalls will need more attention.

The findings also suggest guidance for practitioners. If the application is streaming, runs on constrained devices, or processes very long signals, a state space or recurrent model, or a lightly hybridised one, is likely to be the better choice. If the application depends on retrieving exact details from a large context, such as legal or technical document analysis, attention should be retained in substantial proportion. If compute is not limiting and sequences are short, the optimised Transformer remains hard to beat, and benefits from a mature software ecosystem. Finally, because the fixed-dynamics model performed poorly, input dependence should be considered a requirement for multimodal use, not an enhancement.

Several limitations apply. The models were of moderate size, 350 million parameters, and the behaviour of families may change at larger scales, where reports suggest that gaps narrow but may also widen for recall. Five tasks cannot cover the full diversity of multimodal problems, and generation and instruction following were not studied. Efficiency numbers depend on the hardware and on kernel maturity, which favours attention at present because it has been optimised for years; further engineering of the alternatives may increase their advantage. The tuning budget was equal across architectures, but families with less established recipes may have benefited from more tuning. Finally, the diagnostic probes use synthetic insertions, which may overstate or understate recall in natural documents.

Future work should examine larger scales, the interaction between architecture and multimodal alignment, adaptive hybrids that decide per input how much attention to use, and training methods that improve the recall of fixed-state models, such as learning what to retain with explicit objectives.

8. CONCLUSION

This paper assessed next-generation neural architectures for multimodal sequence modelling under a controlled protocol, with matched parameters, data, and training. The Transformer remained the most accurate single architecture on tasks that need exact recall, while state space, retentive, and recurrent models matched or exceeded it on streaming, sensor, and forecasting tasks and were two to five times faster at long lengths, using a small fraction of the inference memory. Fixed-dynamics models, lacking input dependence, performed clearly worse and are not recommended for multimodal use.

A hybrid with one attention layer for every seven recurrent layers reached 99.1% of Transformer accuracy on average, with 3.4 times the throughput and 71% less memory at 64,000 tokens, and a heavier hybrid matched or surpassed the Transformer on every task. The evidence points to a future in which attention and recurrence are combined deliberately, with attention supplying precise retrieval and recurrence supplying efficient memory, and not to the wholesale replacement of one by the other.

References

1. Pawan Kumar Singh, Scaling Gate-All-Around (GAA) Transistor Architectures for Sub-2nm AI Accelerators Using Atomically Thin 2D Materials, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5713>
2. Pawan Kumar Singh, Energy-Efficient Task Scheduling in Green Cloud Computing Using Neuromorphic Spiking Neural Networks, Vol. 33 No. 08 (2024): JOCAAA, Journal Name: Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5712>
3. A. Mahendra Vardhan and S. Sridhar, "Determining False Positive Analysis of Software Vulnerabilities with Predefined Scan Rules using Random Forest Classifier and Decision Tree Technique," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 622-625, DOI: <https://doi.org/10.1109/ICAC3N56670.2022.10074458>
4. Ronakkumar shah , Cognitive Process Intelligence: Integrating Generative AI and Automated Workflow Discovery, Vol. 33 No. 08 (2024): JOCAAA, Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5676>
5. Ronakkumar shah, research Trends in Smart Workflow Automation: A Bibliometric Analysis of Emerging Themes, Home / Archives / Vol. 33 No. 05 (2024): JOCAAA, Journal of Computational Analysis and Applications, Journal's ISSN: 1521-1398 (Paper),1572-9206 (Online), <https://eudoxuspress.com/index.php/pub/article/view/5675>
6. Dip Bharatbhai Pate, Dynamic vs. Static visualizations: Understanding their use cases in big data analysis, International Journal of Science and Research Archive, 2022, 05(02), 392-395., DOI <https://doi.org/10.30574/ijrsra.2022.5.2.0077>
7. Dip Bharatbhai Patel, Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021, DOI: <https://doi.org/10.30574/ijrsra.2021.2.1.0047>
8. Dip Bharatbhai Patel, Understanding the Impact of Text Analytics on Social Media Sentiment Analysis, Vol. 5 No. 5 (2019) ,September-October, DOI: <https://doi.org/10.32628/CSEIT1936413>
9. Dip Bharatbhai Patel, Real-Time Data Streaming in BI Dashboards: Opportunities and Challenges, Archives / Vol. 5 No. 4 (2019) : July-August, DOI: <https://doi.org/10.32628/CSEIT192142>
10. Dip Bharatbhai Patel, the Role of Data Visualization in Modern Intelligence, Archives / Vol. 2 No. 2 / Research Article, Published, 2020-11-28 DOI: <https://doi.org/10.32996/jcsts.2020.2.2.4>
11. A. Srivastava, "Securing PHI in cloud-based healthcare systems: Challenges, frameworks, and future directions," J. Comput. Anal. Appl., vol. 33, no. 2, Aug. 2024. [Online]. Available: <https://www.eudoxuspress.com/index.php/pub/article/view/4349/3190>
12. A. Srivastava, "AI-assisted cloud migration of healthcare claims data from legacy systems: A metadata-driven framework," Int. J. Commun. Networks Inf. Secur., vol. 14, no. 3, Nov. 2022. [Online]. Available: <https://ijcnis.org/index.php/ijcnis/article/view/8682>
13. A. Srivastava, "Enhancing provider and claims data accuracy using artificial intelligence on cloud-native data platforms," J. Comput. Anal. Appl., vol. 31, no. 4, Nov. 2023. [Online]. Available: <https://eudoxuspress.com/index.php/pub/article/view/4347/3189>
14. A. Srivastava, "A unified framework for secure cloud data management: Integrating governance, metadata intelligence, and privacy controls," J. Inf. Syst. Eng. Manage., vol. 9, no. 2, Apr. 2024. [Online]. Available: https://www.jisem-journal.com/download/45_A_Unified_Framework.pdf
15. A. Srivastava, "Optimizing large-scale data migration for cloud-native architectures," J. Comput. Anal. Appl., vol. 31, no. 2, pp. 652–662, May 2023. [Online]. Available: <https://eudoxuspress.com/index.php/pub/article/view/4603>
16. Abdul Aziz. "Intelligent Health Analytics: Mastering AI in Healthcare Data". Acta Scientiae, vol. 25, no. 4, Aug. 2024, pp. 544-55, <https://www.periodicos.ulbra.org/index.php/acta/article/view/687>. <https://doi.org/10.5281/zenodo.21699886>

17. Event-Aware Multi-Layer Storage Risk Forecasting for Oracle Database Estates Using HAPF, Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 10(4), 2024 ,10.22399/ijcesen.5183
18. Backup Integrity and Recovery Readiness Assessment for High-Availability Databases Raghu Gollapudi, Computer Fraud & Security, Vol. 2023, Issue 8, pp. 35-48, 10.52710/cfs.1031
19. Risk-Controlled Near-Zero-Downtime Oracle Database Migration Using GoldenGate Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 8(3), 2022 ,10.22399/ijcesen.5382
20. An Automated Risk Scoring Framework for SQL Execution Plan Analysis and Performance Manoj Kumar Kagitha, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep), Publication year 2021, Volume 2021, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/970>
21. Manoj Kumar Kagitha, Hybrid Quantum-Classical Resource Scheduling: A Proposed Architecture for Next-Generation Hyperscale Data Centers, Publication year 2024, Volume 2024, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/969>
22. Manoj Kumar Kagitha, Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS), Publication year 2023, Volume 2023, Issue 12, <https://computerfraudsecurity.com/index.php/journal/article/view/971>
23. Manoj Kumar Kagitha, LOAD BALANCING TECHNIQUES IN DISTRIBUTED CLOUD ENVIRONMENTS, (IJESMR - 2017), D.O.I:10.53555/jcr.v6:i1.13380, <https://jcreview.com/archives/volume-6/issue-1/13380>
24. Manoj Kumar Kagitha, AI-DRIVEN OBSERVABILITY FOR HYPERSCALE COLOCATION DATA CENTERS: A CASE STUDY OF LOG ANALYTICS AND ANOMALY DETECTION PIPELINES, Archives / Vol. 27 No. 7 (2019): JOCAAA, <https://www.eudoxuspress.com/index.php/pub/article/view/5053>
25. Manoj Kumar Kagitha, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, <https://www.eudoxuspress.com/index.php/pub/article/view/5054>
26. Bandari, V. V., & Lekkala, H. B. (2024). AI-based operator behavior monitoring and cost optimization using digital traceability in manufacturing systems. Power System Protection and Control, 52(1), 38–45. <https://pspac.info/index.php/dlbh/article/view/342>
27. Venumadhav Vavilala, Shankar Balla (2024, May). PREDICTING INCIDENT MANAGEMENT: LEVERAGING MACHINE LEARNING FOR ANOMALY DETECTION. Power System Protection and Control Scopus Q1 Journal. PSPC. <https://pspac.info/index.php/dlbh/article/view/270>
28. Manikantha Varaprasad Inakollu, (2024, May), FINOPS-Driven Cloud optimization models for enterprise applications, Vol. 52 No. 2 (2024): April-June 2024, 99-110, Power System Protection and Control, ISSN-1674-3415, URL: <https://pspac.info/index.php/dlbh/article/view/283>, DOI: <https://doi.org/10.46121/pspc.52.2.10>
29. Naresh Lokiny. (2022). Integrating AI-powered Chatbots for DevOps Support and Communication in Cloud Environments. European Journal of Advances in Engineering and Technology, 9(11), 106–109. <https://doi.org/10.5281/zenodo.13325989>
30. Naresh Lokiny, (2021), "Disaster Recovery and Business Continuity Planning in DevOps Cloud with AI", International Journal of Science and Research (IJSR), 10(3), 2024-2027. <https://dx.doi.org/10.21275/SR24724151733>, <https://www.ijsr.net/getabstract.php?paperid=SR24724151733>
31. Naresh Lokiny, & Ranganath Nandanampati. (2020). DevSecOps: Integrating Security into DevOps with AI in Cloud. Journal of Scientific and Engineering Research, 7(10), 239–242. <https://doi.org/10.5281/zenodo.13348695>
32. Naresh Lokiny, & Pradip Reddy. (2021). Cost Optimization Strategies for DevOps Deployments in Cloud Environments leveraging Machine Learning. European Journal of Advances in Engineering and Technology, 8(3), 69–72. <https://doi.org/10.5281/zenodo.13325845>
33. Jayanth Para, 6G Internet Technology Cyber Threat Notification & Alert System, Vol. 52 No. 4 (2024): October-December 2024, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/170>, DOI: <https://doi.org/10.46121/pspc.52.4.9>

34. Jayanth Para, AI Based Cloud Computation Observational Method & Process, Vol. 51 No. 4 (2023): October-December 2023, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/171>, DOI: <https://doi.org/10.46121/pspc.51.4.3>
35. Kaleshwar Aryasomayajula, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/294>
36. Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/364>
37. Kaleshwar Aryasomayajula, Micro services: "A Comparative Analysis of Monolithic vs. Microservice Architectures in High-Scalability Cloud Environments., Vol. 51 No. 2 (2023): April-June 2023, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/374>
38. Vikas Suresh Bagora, KERNEL-LEVEL PERFORMANCE OPTIMIZATION FOR CARRIER-GRADE BROADBAND AND HIGH-SPEED NETWORKING SYSTEMS, Vol. 47 No. 1 (2019), Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/359>
39. Vikas Suresh Bagora, SCALABLE 128G FIBRE CHANNEL AND ETHERNET CONVERGENCE FOR NEXT-GENERATION DATA CENTER NETWORKS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/362>
40. Vikas Suresh Bagora, SECURE EMBEDDED NETWORKING ARCHITECTURES USING TRUSTED EXECUTION ENVIRONMENTS IN BROADBAND GATEWAYS, Vol. 52 No. 2 (2024): April-June 2024, Power System Protection and Control, ISSN-1674-3415, <https://pspac.info/index.php/dlbh/article/view/363>
41. Santhosh Chitraju Gopal Varma, The Evolution of Cloud-Native Architectures: Exploring the Synergy between Kubernetes and Microservices, DOI: <https://doi.org/10.63282/3050-9246.IJETCSIT-V11I4P104>, Vol. 1 No. 4 (2020), <https://ijetcsit.org/index.php/ijetcsit/article/view/108>
42. Santhosh Chitraju Gopal Varma, Artificial Intelligence in Cloud Computing: Building Intelligent, Distributed, and Fault-Tolerant Systems, Vol. 3 No. 1 (2022), DOI: <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P105>, <https://ijaibdcms.org/index.php/ijaibdcms/article/view/78>
43. Sunil Karmakar, AI driven Identity & Access Management with Oracle Cloud security, Vol. 30 No. 2 (2022): JOCAAA, <https://eudoxuspress.com/index.php/pub/article/view/4118>
44. Rakshitha Kiran P., Samitha Khaiyum, Aparna R. Palandye, Apoorva S. D., Arjun R., Akshaya S. A., Annapoorna S., Leveraging LLaMA3 and LangChain for Rapid AI Application Development, Vol. 20 No. 10s (2024), <https://doi.org/10.52783/jes.5539>, <https://journal.esrgroups.org/jes/article/view/5539>.