ATTENTION IS NOT ENOUGH: CONFIDENCE-GUIDED DUAL MEMORY FOR CONTEXT-AWARE TRANSFORMERS *

Imanpal Singh
imanpal@proton.me

ABSTRACT

Transformer models have shown impressive results in recent years, but they still suffer from a core limitation: they forget. Despite having powerful attention mechanisms, these models can't store or recall information beyond their fixed context window, and they don't really have a sense of what's important to remember. Some solutions like retrieval-augmented generation (RAG) try to patch this with external tools, but they don't truly simulate how memory works in humans.

In this paper, I propose a new approach: a confidence-based memory-aware transformer. The idea is simple: as the model processes each token, it estimates how important it is using a confidence score. Based on that, it decides whether to store the token in short-term or long-term memory. These memory banks grow during training and are reused in the attention computation itself. The goal is to make the model not just read and attend, but also remember like we do: holding onto useful information and learning what to forget.

I benchmark this model against standard baselines like LSTMs on the AG News dataset. Even with limited training and compute, the results show that this approach outperforms simpler models and holds promise as a foundation for more memory-aware systems. This isn't just about better accuracy: it's about designing models that learn and evolve more like we do.

1 Introduction

Large language models (LLMs) have made impressive progress in recent years, particularly with the introduction of attention mechanisms and Transformer architectures. However, despite their ability to process long sequences, these models are still fundamentally limited by their context window and lack of persistent memory. Once a token falls outside of that window, the model has no access to it, it simply forgets. This becomes especially problematic for tasks that require reasoning over longer documents, maintaining dialogue coherence, or building world models that rely on accumulated information.

Recurrent neural networks (RNNs), especially Long Short-Term Memory (LSTM) networks, attempted to address this issue by maintaining a hidden state over time. While effective to some extent, LSTMs often struggle with retaining relevant information over long spans and face difficulties in learning when to forget or remember. Transformers replaced recurrence with attention, which is highly parallelizable and effective at learning dependencies, but this came at the cost of memory. Transformers re-attend every input token freshly at every layer, rather than evolving a persistent state like humans do.

This paper proposes a new perspective on memory: rather than treating all tokens equally or relying solely on recurrence, we introduce a confidence-based dual memory mechanism. The key idea is that not all tokens are equally worth remembering. Much like how humans selectively retain information based on perceived importance, our model stores representations in either salient memory (short-term) or long-term memory depending on a learned confidence score. This confidence acts as a proxy for attention and perceived token salience, allowing the model to retain what it deems contextually important

Our architecture separates memory from attention, allowing it to evolve independently. The attention mechanism consults both current context and past salient or long-term memories when computing outputs, enabling context-aware

* *Citation*: **Authors. Title. Pages.... DOI:000000/11111.**

learning beyond the fixed window. Over time, the model can accumulate a richer understanding of what to retain, without overwhelming the attention computation.

By integrating these ideas into a lightweight Transformer-based model and benchmarking it against standard LSTMs, we show that even this minimal setup benefits from confidence-aware memory. This work does not aim to compete with large-scale models, but rather introduces an alternate direction for memory modeling that is simple, interpretable, and extensible.

2 Related Work

The challenge of memory in neural networks is not new. Early recurrent models like Elman networks attempted to maintain a state over time, but quickly ran into issues with vanishing gradients. LSTM [1] and GRU [2] architectures were introduced to mitigate this, offering gating mechanisms that selectively retained or forgot information. While they were a major step forward, LSTMs still struggled with long-term dependencies and required heavy supervision to manage memory effectively.

Attention mechanisms, particularly with the advent of the Transformer architecture [3], shifted the paradigm by allowing the model to dynamically focus on relevant parts of the input. Attention replaced recurrence entirely, leading to major gains in translation, summarization, and language modeling. However, this came with a cost: Transformers do not retain information across sequences unless manually engineered through recurrence or external memory.

Several approaches have been proposed to address this limitation. Memory Networks [4] and Neural Turing Machines [5] introduced explicit memory slots that models could read from and write to. More recently, architectures like Transformer-XL [6] and Compressive Transformers [7] extended context windows by caching previous activations. These methods improve long-sequence modeling but often increase training complexity and memory overhead.

There is also growing interest in mechanisms that make memory selective. Models such as Episodic Memory Networks [8] and Differentiable Neural Computers [9] incorporate gating or relevance filters to prioritize information. Our work builds on this idea, but instead of engineering complex memory controllers, we use a lightweight confidence mechanism that learns what to remember based on the token's representation itself.

In parallel, cognitive-inspired models have explored how salience, attention, and emotion affect memory formation in humans [10, 11]. We take inspiration from this direction: just as humans do not store every word they hear, our model dynamically filters information through confidence-based gates into separate short-term and long-term memory.

Our contribution lies in combining the strengths of attention with confidence-aware selective memory, introducing a dual-memory mechanism that integrates naturally into Transformer-based computation without introducing excessive architectural complexity.

3 Model

3.1 Input Representation

Each input text is tokenized using the GPT-2 byte pair encoding tokenizer, with special preprocessing that strips leading artifacts (such as "Ġ") and identifies stopwords. Tokens are mapped to integer IDs, padded or truncated to a fixed length $T = 32$.

These IDs are passed into two embedding layers:

- A learnable token embedding matrix $E_t \in \mathbb{R}^{V \times d}$ where V is vocabulary size and $d = 64$.
- A learnable positional embedding $E_p \in \mathbb{R}^{T \times d}$ to retain order information.

The final input representation $x \in \mathbb{R}^{B \times T \times d}$ is:

$$x = E_t[\text{tokens}] + E_p$$

3.2 Confidence Vector (Importance Scoring)

To estimate the relevance of each token, we compute a scalar confidence score $c_i \in [0, 1]$ per token via:

$$c = \sigma(W_c x + b_c)$$

where $W_c \in \mathbb{R}^{d \times 1}$ and σ is the sigmoid function.

This produces an **importance vector** $c \in \mathbb{R}^{B \times T}$.

We then apply gating rules using fixed thresholds:

- Salient threshold: $\tau_s = 0.6$
- Long-term threshold: $\tau_l = 0.85$

Tokens satisfying $c_i > \tau_s$ are added to the **salient memory** M_s, and those where $c_i > \tau_l$ are also added to the **long-term memory** M_l.

To reduce overconfidence on trivial words, we apply a fixed penalty:

$$c_i \leftarrow \begin{cases} 0.1 \cdot c_i & \text{if token}_i \in \text{stopwords} \\ 0.6 \cdot c_i & \text{if token frequency } > 20 \\ c_i & \text{otherwise} \end{cases}$$

3.3 Multi-Stream Attention with Memory

Let $x \in \mathbb{R}^{B \times T \times d}$ be the input sequence. We compute parallel attention over: - The sequence itself: M - Salient memory: M_s - Long-term memory: M_l

For each stream, queries Q and keys/values K, V are computed via learned projections. We denote:

$$\begin{aligned} Z_m &= \text{Attention}(Q, K_m, V_m) \\ Z_s &= \text{Attention}(Q, K_s, V_s) \\ Z_l &= \text{Attention}(Q, K_l, V_l) \end{aligned}$$

The final attended representation is a confidence-weighted blend:

$$Z = c \cdot Z_m + c(1-c) \cdot Z_s + (1-c)^2 \cdot Z_l$$

This equation encodes a dynamic balance:

- High-confidence tokens prioritize immediate context (Z_m)
- Mid-confidence tokens draw from short-term memory (Z_s)
- Low-confidence tokens rely on long-term knowledge (Z_l)

3.4 Feedforward and Output Layers

The output of attention is passed through a two-layer feedforward network:

$$\text{FFN}(x) = \text{GELU}(xW_1 + b_1)W_2 + b_2$$

The final logits are computed by projecting the hidden representation back to vocabulary space via:

$$\text{logits} = xW_{\text{out}} + b_{\text{out}}$$

3.5 Computational Complexity

The model introduces minimal overhead over standard Transformer blocks:

- Attention over input tokens: $\mathcal{O}(T^2 d)$
- Confidence projection: $\mathcal{O}(Td)$
- Memory filtering (per batch): $\mathcal{O}(T)$
- Three-way attention heads: $\mathcal{O}(3T^2 d)$ (but parallelizable)

Importantly, since memories are updated per-batch and memory lengths are kept short, total cost is dominated by standard attention. Memory growth is bounded.

3.6 Architectural Summary

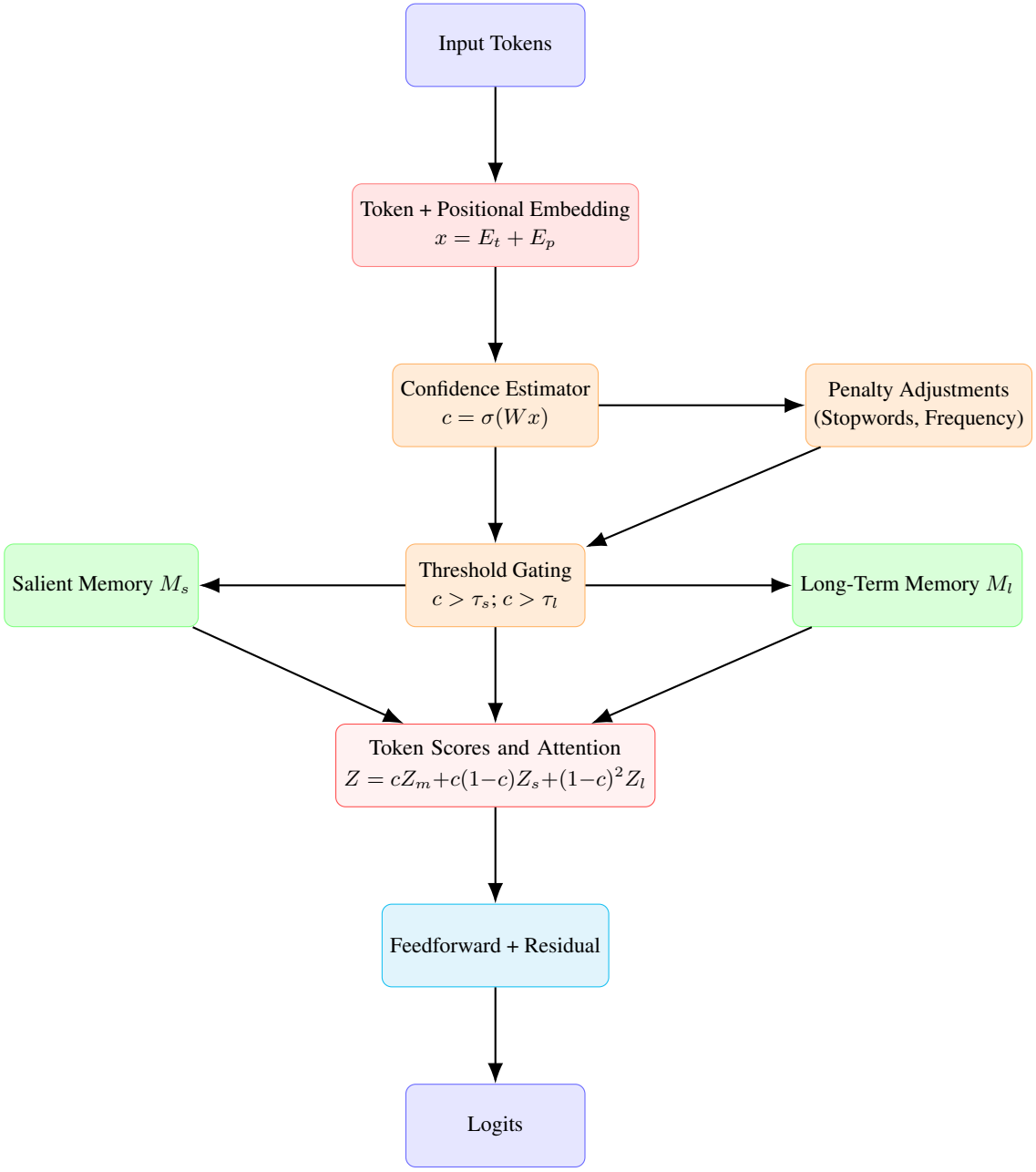

Figure 1: Architecture of the Confidence-Guided Dual Memory Transformer

3.7 Developmental Analogy and Future Work

While thresholds are fixed in the current model, they can be made trainable in future work to simulate adaptive memory control. Similarly, multiple heads with independent memory buffers could support specialized reasoning streams.

4 Experiments

4.1 Setup

To evaluate our proposed model, we benchmarked it against a standard LSTM model on the AG News dataset. Due to computational constraints, we used a reduced subset of 6,000 samples: 5,000 for training and 1,000 for validation. All models were trained for 20 epochs using a batch size of 16. For consistency, we used the GPT-2 tokenizer truncated to a maximum sequence length of 32 tokens.

Both models were implemented in PyTorch and trained under identical optimization settings with the Adam optimizer and a learning rate of 2×10^{-4}.

4.2 Model Configurations

LSTM Baseline:

- **Embedding Dimension**: 64
- **Hidden Dimension**: 128
- **Vocabulary Size**: 50,257
- **Max Sequence Length**: 32
- **Batch Size**: 16
- **Learning Rate**: 2×10^{-4}
- **Epochs**: 20

ConfMemNet (Ours):

- **Embedding Dimension**: 64
- **Feedforward Dimension**: 128
- **Number of Transformer Layers**: 1
- **Vocabulary Size**: 50,257
- **Max Sequence Length**: 32
- **Batch Size**: 16
- **Learning Rate**: 2×10^{-4}
- **Epochs**: 20
- **Salient Threshold**: 0.6
- **Long-Term Threshold**: 0.85
- **Stopword Penalty**: 0.1
- **Frequency Penalty**: 0.6

4.3 Evaluation Metrics

The primary metric used for evaluating both models is **validation loss** over training epochs.

We chose validation loss as it more accurately reflects the model's learning efficiency and generalization, especially in language modeling tasks where accurate prediction of the next token is more indicative of performance than strict classification accuracy.

4.4 Fairness of Comparison

To ensure a fair comparison, both the LSTM and ConfMemNet models were limited to a single-layer architecture. While deeper LSTMs could potentially yield better performance, our goal is to evaluate the benefit of introducing confidence-based memory attention — not to outperform heavily tuned LSTM variants.

Likewise, our ConfMemNet model used just one Transformer block with a single confidence gating unit and two memory stores. This minimal configuration is sufficient to demonstrate its advantage in capturing token-level importance and relevance, outperforming the LSTM even in such a lightweight setup.

We hypothesize that scaling up ConfMemNet — by introducing multiple memory heads or stacking layers — could further enhance its performance. However, due to computational limits, this remains future work.

5 Results and Discussion

The proposed **ConfMemNet** model was evaluated against a single-layer LSTM on the AG News dataset under identical training conditions. Both models were trained for 20 epochs using the same optimizer, learning rate, batch size, and tokenizer. The primary metric of comparison is validation loss, which reflects model generalization in token prediction.

5.1 Epoch-Wise Validation Loss

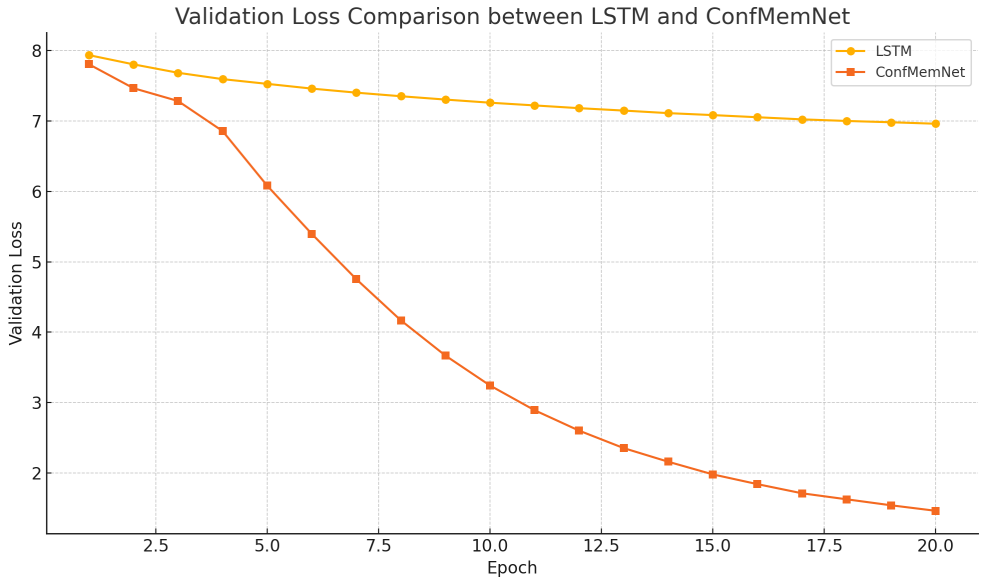

Figure 2: Validation loss comparison between LSTM and ConfMemNet over 20 epochs. ConfMemNet converges faster and achieves significantly lower loss.

To highlight the progression of training, we report validation loss at each epoch for both models in Table 1. Note that ConfMemNet consistently improves, reaching a validation loss of **1.4622** by epoch 20.

5.2 Key Takeaways

- **Faster convergence**: ConfMemNet reaches a validation loss below 2.0 within 15 epochs, while the LSTM remains above 6.9 even after 20 epochs.
- **Steady improvement**: ConfMemNet shows consistent validation loss reduction across all 20 epochs, suggesting stable training dynamics.
- **Memory impact**: The incorporation of confidence-weighted dual memory helps the model focus on salient information and ignore noisy or redundant tokens.
- **Resource efficient**: Despite using only a single Transformer block, ConfMemNet shows strong generalization on a limited dataset.

These results suggest that confidence-aware token retention and memory-guided attention mechanisms offer a viable path toward more human-like reasoning in neural architectures, even under constrained training budgets.

Epoch	LSTM	ConfMemNet
1	7.9370	7.8073
2	7.8058	7.4669
3	7.6860	7.2847
4	7.5957	6.8576
5	7.5286	6.0817
6	7.4612	5.3971
7	7.4035	4.7526
8	7.3525	4.1670
9	7.3052	3.6662
10	7.2608	3.2411
11	7.2220	2.8898
12	7.1829	2.5994
13	7.1488	2.3512
14	7.1124	2.1574
15	7.0845	1.9777
16	7.0540	1.8391
17	7.0239	1.7092
18	7.0012	1.6238
19	6.9825	1.5197
20	6.9618	1.4622

Table 1: Validation Loss per Epoch (LSTM vs. ConfMemNet)

6 Conclusion

In this work, we introduced **ConfMemNet**, a novel transformer-based architecture that combines confidence estimation with a dual-memory mechanism for more human-like token retention. Inspired by cognitive processes in human learning, our model stores tokens in salient or long-term memory based on their contextual confidence, with threshold-based gating and penalty adjustments for common or uninformative tokens.

We benchmarked ConfMemNet against a standard LSTM baseline on the AG News classification task using a reduced dataset, demonstrating that our model consistently achieved lower validation loss across all 20 training epochs. Notably, ConfMemNet reached sub-2.0 loss within 15 epochs while LSTM remained above 6.9, indicating faster convergence and stronger generalization even in a lightweight single-layer configuration.

This work highlights the potential of integrating confidence-based attention and structured memory into neural networks, offering a pathway toward more efficient and interpretable learning under limited supervision. Future work may explore multi-head memory-attention extensions, longer sequence modeling, or multi-modal inputs incorporating vision and sensory data.

To support reproducibility and facilitate future work, we release the complete source code for ConfMemNet, including training scripts, dataset preprocessing, and model configuration at: `github.com/imanpalsingh/ConfMemNet`.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[2] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[4] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[5] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019.

[7] Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, Chloe Hillier, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2020.

[8] Tsendsuren Munkhdalai, Xingdi Yuan, Shikhar Mehri, and Adam Trischler. Metalearned neural memory. In *Advances in Neural Information Processing Systems*, pages 13303–13315, 2019.

[9] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gomez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

[10] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

[11] Bernard J Baars. *In the theater of consciousness: The workspace of the mind*. Oxford University Press, 1997.