Investigating Architectural Placement of Hopfield Memory Layers in CHopT for Improved Fluency

Benjamin Chauhan
Duke University
benjamin.chauhan@duke.edu

Project Report

https://github.com/Meeeee6623/CHAD

Abstract
The original CHopT (Continual Hopfield Transformers) architecture demonstrated that augmenting LLMs like Llama 3.2 3B with a post-transformer Hopfield memory layer enables cross-inference factual recall. However, this end-of-network placement often resulted in grammatically poor output, as the retrieved memory bypassed the main transformer blocks responsible for language modeling. This report details subsequent experiments I conducted to test the hypothesis that integrating Hopfield memory layers *earlier* within the Llama 3.2 3B stack could improve output fluency by allowing the transformer blocks to process the retrieved information. I investigated two alternative strategies: a "pre-middle" configuration (layers after embeddings and mid-stack) and a "deep" configuration (layers before every transformer block). While the pre-middle setup yielded quantitative ROUGE-1 scores comparable to the baseline, qualitative results showed a significant *degradation* in generation fluency, contradicting the initial hypothesis. The deep integration approach failed to converge entirely during LoRA-based fine-tuning. These findings suggest that under constrained LoRA fine-tuning, earlier integration of memory does not improve fluency and can severely destabilize the model, highlighting the difficulty of tightly coupling external memory with pretrained transformer dynamics without more extensive training or more careful integration methods.

1 Introduction

Transformer-based Large Language Models (LLMs) [9] excel at sequence modeling but are inherently stateless between inference calls, hindering their ability to build upon past interactions [1, 2]. While Retrieval-Augmented Generation (RAG) [5] provides access to external knowledge, it doesn't create an evolving internal memory state.

The CHopT (Continual Hopfield Transformers) project, previously undertaken by Peter Liu and Benjamin Chauhan, offered a promising step by integrating a learnable, gated Hopfield memory layer [8] into the Llama 3.2 architecture [7]. The initial work placed this memory layer *after* the final Llama transformer block. This configuration demonstrated successful factual recall across inference boundaries on the NarrativeQA benchmark [4]. However, a key limitation emerged: because the retrieved memory ($\mathbf{r}$) was fused with the Llama output ($\mathbf{h}_{\text{llama}}$) *after* all transformer processing, the sophisticated language modeling capabilities of the pretrained Llama model did not act upon the retrieved facts. This often resulted in outputs that contained correct factual elements but suffered from poor grammar and fluency (see Table 2). Essentially, the bulk of the model remained unaware of the memory content during generation.

This observation motivated the central hypothesis of the work presented in this report: Could integrating the Hopfield memory cells *earlier* in the transformer architecture improve the grammat-

ical fluency of memory-driven outputs? The rationale was that if memory retrieval happened before or between transformer blocks, the subsequent layers could process the combined contextual and retrieved information, leveraging Llama's pretrained language abilities to generate more coherent and grammatically correct sentences incorporating the recalled facts.

To test this hypothesis, I explored alternative architectural placements for the Hopfield memory layer within Llama 3.2 3B, comparing them against the original end-of-network configuration using the same cross-inference NarrativeQA task and LoRA fine-tuning methodology. This report focuses on empirical evaluation of these earlier placements (Pre-Middle and Deep) and the analysis of whether they successfully addressed the fluency limitations of the baseline while maintaining recall, particularly under the constraint of LoRA fine-tuning.

2 Background: Modern Hopfield Networks & Attention

Understanding the CHopT memory mechanism requires familiarity with modern continuous-state Hopfield networks [8]. These networks act as associative memories by minimizing an energy function. Given a state (query) ξ and stored patterns X, the network iteratively updates the state to converge towards an energy minimum, which often corresponds to one or a combination of the stored patterns most similar to the initial query.

Crucially, the core update rule derived from this energy minimization, $\xi_{\text{new}} = X \cdot \text{softmax}(\beta X^\top \xi_{\text{old}})$, is mathematically equivalent to the scaled dot-product attention mechanism central to Transformers [9]. Here, the query ξ attends to the stored patterns X, which serve as both keys and values (or are projected into keys and values). The parameter β controls the sharpness or temperature of the softmax, influencing the focus of the retrieval. This Hopfield-Attention equivalence [8] provides a powerful and theoretically grounded framework for implementing a high-capacity, content-addressable, and differentiable memory system within a neural network.

3 Recap: The Baseline CHopT Architecture and its Limitations

To understand the placement experiments, we first recap the core components and observed limitations of the original CHopT architecture, where the memory layer was placed *after* the final Llama transformer block (Figure 1).

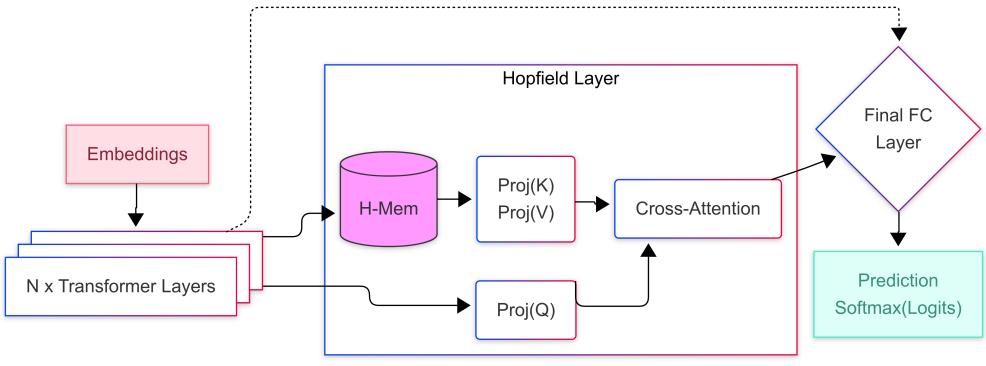

Figure 1: Original CHopT architecture overview (Liu and Chauhan). The Hopfield Memory layer (H-Mem) processed the final Llama state ($\mathbf{h}_{\text{llama}}$). Retrieved memory was fused *after* Llama's processing, before the final prediction layer.

3.1 Hopfield-Attention Associative Memory Layer

The core memory component is the HopfieldMemoryLayer (detailed in Figure 2).

- **Internal State (P):** A learnable ‘nn.Parameter‘ tensor representing persistent memory patterns (H, M, D_h).
- **Retrieval Process:** Incoming hidden state (final $\mathbf{h}_{\text{llama}}$) projected to Queries ($\mathbf{Q}$). Internal memory $\mathbf{P}$ projected to Keys ($\mathbf{K}$) and Values ($\mathbf{V}$).
- **Association & Aggregation:** Scaled dot-product attention $A = \text{softmax}(\frac{\beta \mathbf{Q}\mathbf{K}^\top}{\sqrt{D_h}})$ retrieves memory $\mathbf{r} = A\mathbf{V}$. Iterative retrieval ($N = 2$) often improved recall.
- **Fusion:** Retrieved memory $\mathbf{r}$ is normalized and additively fused with the *final* Llama state: $\mathbf{h}_{\text{llama}} + \text{norm}(\mathbf{r})$. This fused state goes directly to the final layer norm and output projection.

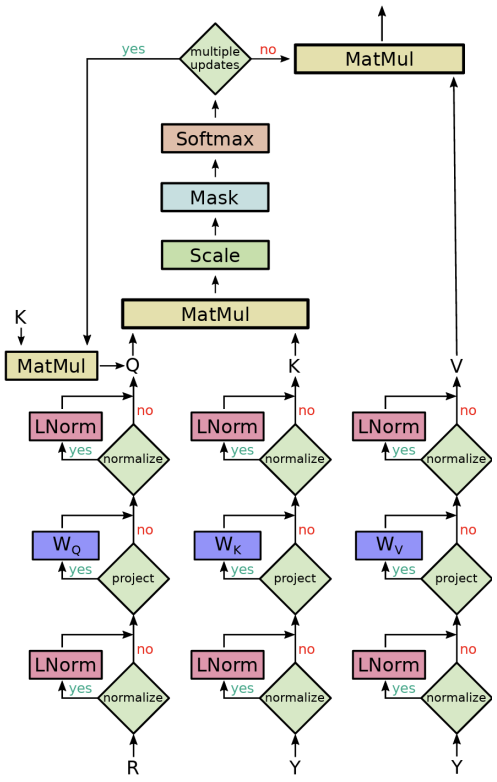

Figure 2: Original Hopfield layer retrieval mechanism detail, based on attention [9]. Input queries (from final Llama state) and memory patterns are projected; attention retrieves relevant memory values.

3.2 Differentiable Gated Memory Update

A differentiable gate $\mathbf{G}$ modulated memory updates $\mathbf{P} \mathrel{+}= \eta_{\text{mem}} \cdot \mathbf{G} \cdot (\mathbf{t} - \mathbf{P})$ within the computation graph, allowing end-to-end learning of plasticity control.

3.3 Observed Limitation: Late Fusion and Poor Fluency

While the baseline end-of-network CHopT demonstrated factual recall (Table 1), a significant qualitative issue was the often poor grammatical structure and fluency of the generated answers con-

taining recalled information (Table 2). The likely cause is the *late fusion* strategy. By adding the retrieved memory $\mathbf{r}$ only after the entire Llama transformer stack had finished processing, the powerful sequence modeling and language generation capabilities inherent in the pretrained Llama blocks were bypassed. The model could retrieve facts but struggled to integrate them smoothly into well-formed sentences. This limitation provided the primary motivation for exploring earlier memory integration points.

4 Experimental Setup

4.1 Task and Dataset: Cross-Inference NarrativeQA

To test the hypothesis regarding improved fluency via earlier integration, I used the same cross-inference paradigm on the NarrativeQA dataset [4] as the original CHopT work:

1. **Context Encoding Phase:** Narrative summaries processed sequentially during fine-tuning, allowing memory updates via the gated mechanism (§3.2).
2. **Question Answering Phase (Cross-Inference):** Questions presented without context; answers generated solely based on retrieved memory (§3.1). Memory updates disabled.

This setup isolates the model's reliance on its internal state and allows evaluation of both recall and generation quality.

4.2 Model Configurations and Training

- **Base Model:** Llama 3.2 3B Instruct [7].
- **Baseline CHopT (End-of-Network):** As described in Section 3, using best parameters from prior work ($H = 24, M = 512, D_h = 128, N = 2$).
- **Pre-Middle CHopT:** Two Hopfield layers ($H = 24, M = 96, D_h = 128$ each) placed after embeddings and after block 14.
- **Deep CHopT:** Small Hopfield layers ($H = 16, M = 24, D_h = 128$) placed before each of the 28 Llama blocks.
- **Fine-tuning Strategy (Consistent):** LoRA [3] (r=16, $\alpha = 32$) on base Llama layers. Hopfield parameters fully trained. AdamW 8-bit optimizer (Base LR 2×10^{-5}, Hopfield LR $\times 5$), cosine decay, grad clipping (1.0), gradient checkpointing. 2 epochs on NarrativeQA train split, effective batch size 64.

4.3 Evaluation Metrics

Primary quantitative metric: ROUGE-1 F1 score [6] for factual recall. Primary qualitative metric: Assessment of grammatical fluency and coherence of generated answers, specifically looking for improvements over the baseline. Training stability was also monitored.

5 Results: Architectural Placement Experiments

5.1 Baseline Performance Recap

The original end-of-network CHopT ($M = 512, N = 2$) achieved a ROUGE-1 F1 of 0.11446 (Table 1). Qualitatively, it recalled facts but often with flawed grammar (Table 2), motivating the exploration of earlier integration.

Table 1: Original CHopT ROUGE-1 F1 Scores (Liu and Chauhan) - Hopfield Layer After Final Block

Memory Slots (M per Head)	Update Steps (N)	ROUGE-1
768	2	0.083 10
512	**2**	**0.114 46**
128	2	0.096 00
128	1	0.064 61

Table 2: Original Qualitative Examples (Liu and Chauhan) - $M = 512, N = 2$, End-of-Network. Note the weak factual recall and awkward phrasing (e.g., "He it destroyed.", "She was prostitute.").

Question / Ground Truth	CHopT Output	Base Llama Output
What does Mark do with his radio station? **GT:** He dismantles it... makes mobile transmitter	`He it destroyed.`	`I don't have enough info...`
What did Mrs. Warren previously do for work? **GT:** Prostitution.	`She was prostitute.`	`I'm not sure...`
Why did McLeach throw Cody's backpack...? **GT:** To make people think he had died	`He wanted kill Cody.`	`(not tested)`
Who commits suicide? **GT:** Malcolm	`Malcolm.`	`Suicide is complex... [generic]`
Who is Lapp? **GT:** A Frenchman revealed to be a spy	`He a spy.`	`Lapp is a common last name...`
What does the radio show cause? **GT:** Trouble (FCC investigation)	`A panic.`	`There are multiple talk shows...`
What does Paige jam into her microwave? **GT:** Her medals and accolades	`A can of.`	`I don't have enough context...`

5.2 Architectural Placement Results

I evaluated the two alternative placement strategies to test the hypothesis that earlier integration improves fluency.

1. **Pre-Middle Configuration:** Two Hopfield layers (after embeddings, after block 14).

   Finding: This configuration trained stably using LoRA, achieving a ROUGE-1 F1 score of **0.105849**, quantitatively similar to the baseline. However, contrary to the hypothesis, qualitative analysis (Table 3) revealed that generation fluency was significantly *worse* than

the baseline. Answers were highly fragmented and often nonsensical, indicating that processing retrieved memory through subsequent transformer blocks did not improve grammatical integration in this setup, and likely disrupted the pretrained model significantly.

2. **Deep (All Layers) Configuration:** Small Hopfield layers before each Llama block.

   Finding: This configuration **failed to converge** during LoRA fine-tuning, with ROUGE-1 scores remaining at 0. The model produced no meaningful output (Table 4), rendering it impossible to evaluate the fluency hypothesis. The complete instability suggests this level of integration severely disrupted the pretrained model's function under LoRA.

Table 3: Qualitative Examples: Pre-Middle Configuration Recall (Cross-Inference QA). Output is fragmented and nonsensical, showing degraded fluency compared to baseline, despite similar ROUGE-1.

Question	Ground Truth	Pre-Middle Output
Who is Mark Hunter?	He is a high school student in Phoenix.	`is A`
Where does this radio station take place?	It takes place in Mark's parents basement.	`is in want to to a to what`
Why do more students tune into Mark's show?	Mark talks about what goes on at school and in the community.	`is to?assistant  was to to's:`
Who commits suicide?	Malcolm.	`M is 's?TO  DOES is to what`
What does Paige jam into her microwave?	She jams her medals and accolades.	`assistant  is?A  does say is to to what`

Table 4: Qualitative Examples: Deep (All Layers) Configuration Recall (Cross-Inference QA). Complete failure to generate meaningful text, consistent with convergence failure.

Question	Ground Truth	Deep Output
Who is Mark Hunter?	He is a high school student in Phoenix.	`?`
Where does this radio station take place?	It takes place in Mark's parents basement.	`the?`
Who commits suicide?	Malcolm.	`?`
Who is called in to investigate Mark's radio station?	The FCC	`<EOS Token> (no output)`
When Maskull wakes up in Tormance, how many of his friends are with him?	None	`5 the of, is's?`

5.3 Analysis of Placement Results

The results from the placement experiments challenge the initial hypothesis that earlier memory integration would improve fluency under LoRA fine-tuning.

The Pre-Middle configuration achieved comparable ROUGE-1 scores to the baseline, suggesting that factual elements could be stored and retrieved from earlier representational stages. However, the significant drop in qualitative fluency (Table 3) indicates that allowing subsequent transformer blocks to process the fused memory representations did not lead to better grammatical integration, and likely only shows that the model learned to produce tokens that overlap with the output, not necessarily good responses. Instead, it seems the fusion process itself, or the LoRA-adapted layers' inability to handle the modified input distribution at these earlier stages, severely disrupted the generation process, leading to fragmented and incoherent output. The hypothesis that later blocks would naturally smooth the output was not supported.

The Deep configuration's complete failure to converge and generate meaningful text (Table 4) further underscores the difficulty of deep memory integration with limited adaptation. The potential reasons remain:

- **Disruption to Pretrained Dynamics:** Ubiquitous insertion likely breaks fundamental processing pathways learned during pretraining.
- **LoRA Limitations:** Insufficient adaptation capacity to reconcile the base model with the pervasive architectural changes.

This experiment could not even test the fluency hypothesis due to the training instability.

6 Discussion

The investigation into architectural placement yielded critical insights, primarily contradicting the initial hypothesis regarding fluency improvement via earlier memory integration under LoRA. While the baseline end-of-network CHopT showed promise for recall despite fluency issues potentially caused by late fusion, attempting to fix this by moving memory earlier did not succeed with the limited adaptation afforded by LoRA.

The Pre-Middle configuration demonstrated that factual recall (measured by ROUGE) can be maintained with earlier integration, but at the cost of severely degraded linguistic coherence. This suggests that simply feeding retrieved memory into later transformer blocks isn't sufficient to ensure fluent integration. The interaction is complex; the fusion might disrupt the representational flow expected by the pretrained layers, or the LoRA adaptation might not be sufficient to teach the later layers how to properly incorporate the memory content into grammatical structures. It might be that the specific additive fusion method used is particularly disruptive when applied earlier in the network.

The catastrophic failure of the Deep configuration highlights the fragility of pretrained models when subjected to pervasive architectural changes, especially when fine-tuning is constrained (e.g., by LoRA). The disruption likely overwhelmed the model's ability to adapt, preventing any meaningful learning or generation.

These results suggest that for integrating persistent memory using methods like CHopT with limited fine-tuning (LoRA): 1. End-of-network placement, despite its fluency limitations, appears to be the most stable and effective configuration tested so far, balancing recall and basic generation capability. 2. Earlier integration (like Pre-Middle) might preserve some recall but severely harms fluency, indicating that the problem is not simply about *when* fusion occurs but likely *how* it occurs

and how well the network adapts to it. 3. Very deep integration (like Deep) is likely infeasible without more complete pre-training or significantly different integration/adaptation strategies.

The original baseline's fluency problem remains an open challenge. It might require different fusion techniques (not just additive), more sophisticated gating, or adaptation methods beyond LoRA even for the end-of-network placement. Addressing fluency seems more complex than just changing the integration point.

7 Future Work

To achieve fluent memory integration without relying solely on end-of-network placement, future work should explore more subtle approaches. Some promising directions might involve testing sparser deep placements, like inserting memory layers at moderate intervals (e.g., every 4-8 blocks). Crucially, this likely needs to be combined with refined fusion methods beyond simple addition; techniques like learned gating between hidden states and memory or projecting concatenated representations might integrate retrieved information more smoothly into the transformer's processing flow. An even tighter integration could involve modifying the base model components themselves, such as adapting the attention or feed-forward mechanisms to directly interact with memory patterns, potentially being less disruptive than inserting entirely separate layers. Success in these deeper integrations may also depend on enhanced memory control via improved gating/update rules and likely requires alternative adaptation strategies more powerful than LoRA, such as full fine-tuning or more targeted parameter updates, to effectively train the combined architecture.

8 Conclusion

This report detailed experiments I conducted to test the hypothesis that integrating CHopT's Hopfield memory earlier within the Llama 3.2 3B architecture could improve output fluency compared to the original end-of-network placement, which suffered from grammatical issues potentially due to late memory fusion. Contrary to this hypothesis, under LoRA fine-tuning, placing memory layers earlier (Pre-Middle configuration) maintained comparable quantitative recall (ROUGE-1) but resulted in significantly worse generation quality. A deeper integration (Deep configuration) failed to converge entirely. These results highlight that, with limited adaptation like LoRA, earlier memory fusion does not automatically solve fluency problems and can severely destabilize the model or degrade output quality. The interaction between integrated memory and pretrained transformer blocks is complex, and naive earlier integration appears disruptive. While integrated memory remains promising, achieving both factual recall and fluent generation, especially via deeper integration, likely requires more sophisticated fusion methods, adaptation techniques beyond LoRA, or potentially full fine-tuning. For now, the end-of-network placement remains the most stable, albeit imperfect, configuration identified for CHopT under LoRA constraints.

References

[1] Alexander A Bulatov, Yuri A Kurenkov, and Varvara R Logacheva. Recurrent memory transformer. *arXiv preprint arXiv:2207.06881*, 2022.

[2] 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. Association for Computational Linguistics, 2019. URL https://doi.org/10.18653/v1/P19-1285.

[3] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[4] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. In *Transactions of the Association for Computational Linguistics*, volume 6, pages 317–328. MIT Press, 2018. URL https://doi.org/10.1162/tacl_a_00023.

[5] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Gustavo Nogueira, Hubert Rinh, Wen-tau Chen, Saujas Wang, Sameer Singh, Sebastian Riedel, and Wen-tau Yih. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33*, pages 9459–9474. Curran Associates, Inc., 2020.

[6] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W04/W04-1013.

[7] AI @ Meta. Llama 3.2: The Next Generation of Open Foundational Models. https://ai.meta.com/blog/llama-3-2/, October .

[8] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=pVBVKHU6BEr.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.