HMST: Hierarchical Memory-State Transformer for Hallucination Mitigation in Long-Context Language Models

Anonymous Authors
Institution Name
email@institution.edu

Abstract

Large Language Models (LLMs) exhibit a fundamental trade-off between long-context retention, computational efficiency, and hallucination rates. Existing approaches either scale quadratically with context length (dense transformers), compress information lossy (state-space models), or lack contextual statefulness (retrieval-augmented generation). We introduce **HMST** (Hierarchical Memory-State Transformer), a novel hybrid architecture that integrates a sparse Mixture-of-Experts (MoE) backbone with a meta-controlled three-tier memory system. A lightweight *Meta-Controller*, optimized via reinforcement learning (PPO), dynamically routes queries between: (1) immediate self-attention, (2) episodic state-space compression (Mamba-based SSM), and (3) semantic vector retrieval (FAISS). An optional critic model provides hallucination detection through output verification. On long-context benchmarks, HMST demonstrates superior hallucination resistance (15% reduction in error rate on TruthfulQA) while maintaining sub-linear computational complexity. Our architecture achieves 2.3× faster inference than dense transformers on 16K-token contexts with 12B total parameters (2B active).

1 Introduction

The success of Large Language Models [1, 2, 3] has been accompanied by persistent challenges in long-context reasoning and factual accuracy. Current architectures face three fundamental limitations:

(1) Quadratic Scaling: Self-attention mechanisms in transformers scale as $O(L^2)$ with sequence length L, imposing prohibitive computational costs for contexts exceeding 8K tokens [4, 5].

(2) Lost-in-the-Middle: Dense attention struggles to maintain uniform retrieval accuracy across extended contexts, exhibiting U-shaped performance curves where mid-sequence information is preferentially forgotten [6].

(3) Hallucination Persistence: Despite advances in pre-training scale and alignment, LLMs continue to generate plausible but factually incorrect outputs, particularly when forced to extrapolate beyond training distributions [7, 8].

Existing mitigation strategies exhibit complementary weaknesses. *Retrieval-Augmented Generation* (RAG) [9] offloads long-term memory to external datastores but lacks stateful compression of conversation context. *State Space Models* (SSMs) [10, 11] achieve linear complexity but sacrifice the ability to perform targeted retrieval over historical context. *Sparse attention* patterns [12, 13] reduce compute but maintain quadratic worst-case complexity.

1.1 Contributions

We propose HMST, a hierarchical architecture that synergistically combines three memory tiers under unified RL-based control:

1. **Sparse MoE Backbone:** A 12B-parameter base model with 8 experts and top-2 routing, activating only 2B parameters per token for efficient processing.

2. **Three-Tier Memory Hierarchy:**

   - *L1 (Working Memory):* Standard self-attention over immediate context (∼2K tokens).
   - *L2 (Episodic Memory):* Input-dependent Mamba SSM compressing recent interactions into 256D states (8K effective tokens).

- *L3 (Semantic Memory):* FAISS vector index enabling sub-logarithmic retrieval over 1M+ historical entries.

3. **RL-Optimized Meta-Controller:** A lightweight 6-layer transformer trained via PPO to dynamically route queries based on complexity, uncertainty, and compute budget.

4. **Integrated Critic Model:** A 1B-parameter verification network that detects hallucinations through logical consistency checking.

Our experiments demonstrate 15% hallucination reduction on TruthfulQA, 2.3× inference speedup on long contexts, and superior scaling properties compared to dense baselines.

2 Related Work

Sparse Models & Mixture of Experts. Switch Transformer [14] and Mixtral [15] demonstrate that sparse expert activation maintains model quality while reducing per-token compute. Our MoE backbone follows top-K routing principles but integrates with memory-augmented components rather than serving as the sole architecture.

State Space Models. Structured SSMs (S4 [10], Mamba [11], Jamba [16]) achieve linear complexity through selective state compression. While Jamba explores SSM-attention hybrids at the layer level, HMST implements hierarchical memory with explicit meta-control for routing decisions.

Memory-Augmented Networks. Neural Turing Machines [17] and Memory Networks [18] pioneered external memory interfaces. Modern RAG systems [9, 19] combine dense retrievers with generative models. HMST extends this paradigm with multi-tier memory and learned access policies.

Hallucination Detection. Recent work addresses factual errors through self-consistency checking [20], retrieval verification [21], and uncertainty quantification [22]. Our critic model explicitly conditions on retrieved evidence for calibrated confidence estimation.

3 Architecture

3.1 Notation and Problem Setup

Given an input sequence $\mathbf{x} = (x_1, \ldots, x_L)$ where $x_i \in \mathcal{V}$ (vocabulary), we seek to model the conditional distribution $p(x_{t+1}|x_{\leq t})$ while maintaining three objectives:

$$\text{Minimize:} \quad \mathcal{L}_{LM} = -\sum_{t=1}^{L} \log p(x_t|x_{<t}) \quad \text{(Language Modeling)} \tag{1}$$

$$\text{Minimize:} \quad \mathcal{C}_{compute}(x) \quad \text{(Computational Cost)} \tag{2}$$

$$\text{Maximize:} \quad \mathcal{Q}_{factual}(x) \quad \text{(Factual Accuracy)} \tag{3}$$

Let d_{model} denote embedding dimension, $N = 8$ the number of experts, and $K = 2$ the activation sparsity.

3.2 Base Mixture-of-Experts Layer

Each MoE layer consists of N expert networks $\{E_i\}_{i=1}^{N}$ where $E_i : \mathbb{R}^{d_{model}} \to \mathbb{R}^{d_{model}}$ is a two-layer feed-forward network:

$$E_i(\mathbf{x}) = W_{2,i}\,\text{GELU}(W_{1,i}\mathbf{x} + \mathbf{b}_{1,i}) + \mathbf{b}_{2,i} \tag{4}$$

where GELU is the Gaussian Error Linear Unit activation [23].

The gating network $G(\mathbf{x})$ computes routing probabilities:

$$\mathbf{h} = W_g\mathbf{x} \in \mathbb{R}^N \tag{5}$$

$$\mathbf{p} = \text{Softmax}(\mathbf{h}) \tag{6}$$

Top-K routing selects the expert subset $\mathcal{T} = \text{TopK}(\mathbf{p}, K = 2)$ and renormalizes:

$$\mathbf{y} = \sum_{i \in \mathcal{T}} \frac{p_i}{\sum_{j \in \mathcal{T}} p_j} E_i(\mathbf{x}) \tag{7}$$

Load Balancing. To prevent expert collapse, we minimize the auxiliary loss:

$$\mathcal{L}_{bal} = \alpha \cdot N \sum_{i=1}^{N} f_i \cdot \bar{p}_i \tag{8}$$

where f_i is the fraction of tokens routed to expert i in the batch, $\bar{p}_i = \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} p_i(x)$ is the average routing probability across the batch, and $\alpha = 0.01$ is a hyperparameter.

3.3 Meta-Controller

The Meta-Controller π_θ is a 6-layer transformer (150M parameters) that observes query embedding $\mathbf{e}_q$ and state summary $\mathbf{s}_t$:

$$\mathbf{s}_t = [\text{pos}_t, \text{uncert}_t, \text{len}_t, \text{conf}_t] \tag{9}$$

where:

- pos_t: Positional encoding
- uncert_t: Epistemic uncertainty from prior predictions
- len_t: Current context length (normalized)
- conf_t: Moving average of model confidence scores

The controller outputs five decision gates via separate linear heads:

$$g_{exit} = \sigma(W_{exit}\mathbf{z}_t) \quad \text{(Early Exit)} \tag{10}$$

$$g_{epi} = \sigma(W_{epi}\mathbf{z}_t) \quad \text{(Episodic Access)} \tag{11}$$

$$g_{sem} = \sigma(W_{sem}\mathbf{z}_t) \quad \text{(Semantic Access)} \tag{12}$$

$$g_{ver} = \sigma(W_{ver}\mathbf{z}_t) \quad \text{(Verification)} \tag{13}$$

$$\boldsymbol{\pi}_{exp} = \text{Softmax}(W_{exp}\mathbf{z}_t) \quad \text{(Expert Weights)} \tag{14}$$

where $\mathbf{z}_t = \text{TransformerEncoder}([\mathbf{e}_q; \mathbf{s}_t])$ and σ is the sigmoid function.

Binary decisions are made via thresholding: $a_* = \mathbb{1}[g_* > \tau]$ with $\tau = 0.5$. During training, we use Gumbel-Softmax [24] for differentiability.

3.4 Episodic Memory: Selective State Space Model

The L2 episodic memory implements a selective SSM following the Mamba architecture [11]. For input sequence $\{\mathbf{x}_t\}$, the continuous-time state-space equations are:

$$\mathbf{h}'(t) = \mathbf{A}\mathbf{h}(t) + \mathbf{B}\mathbf{x}(t) \tag{15}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{h}(t) \tag{16}$$

Selective Discretization. We compute input-dependent time-scales:

$$\Delta_t = \text{Softplus}(W_\Delta \mathbf{x}_t + \mathbf{b}_\Delta) \tag{17}$$

The continuous parameters are discretized via zero-order hold:

$$\bar{\mathbf{A}}_t = \exp(\Delta_t \mathbf{A}) \tag{18}$$

$$\bar{\mathbf{B}}_t = (\Delta_t \mathbf{A})^{-1}(\exp(\Delta_t \mathbf{A}) - \mathbf{I})\Delta_t \mathbf{B} \tag{19}$$

The discrete recurrence becomes:

$$\mathbf{h}_t = \bar{\mathbf{A}}_t \mathbf{h}_{t-1} + \bar{\mathbf{B}}_t \mathbf{x}_t \tag{20}$$

$$\mathbf{y}_t = \mathbf{C}_t \mathbf{h}_t + \mathbf{D}\mathbf{x}_t \tag{21}$$

where $\mathbf{B}_t = W_B \mathbf{x}_t$ and $\mathbf{C}_t = W_C \mathbf{x}_t$ provide input-dependent selectivity.

Efficiency. The recurrence processes 8K tokens into a fixed 256D state vector $\mathbf{h}_t$, enabling $O(1)$ memory footprint for retrieval.

3.5 Semantic Memory: Vector Index Retrieval

The L3 semantic memory uses a FAISS index [25] with IVF (Inverted File) and PQ (Product Quantization):

- **Indexing:** Historical embeddings $\{\mathbf{v}_i\}_{i=1}^M$ are clustered into $n_{clusters} = \sqrt{M}$ Voronoi cells.
- **Quantization:** Each 1024D vector is compressed into 64 bytes via 8-byte sub-vectors.

For query $\mathbf{q}$, retrieval proceeds as:

$$\mathcal{R}_k(\mathbf{q}) = \underset{\mathbf{v}\in\mathcal{I}}{\text{TopK}} \left(\frac{\mathbf{q}^\top \mathbf{v}}{\|\mathbf{q}\|\|\mathbf{v}\|} \right) \tag{22}$$

with sub-linear complexity $O(\sqrt{M})$ due to IVF clustering with $\approx \sqrt{M}$ clusters.

Background Consolidation. Importance scores $I(\mathbf{h}_t)$ are computed for episodic states:

$$I(\mathbf{h}_t) = \alpha \cdot \text{novelty}(\mathbf{h}_t) + (1-\alpha) \cdot \text{frequency}(\mathbf{h}_t) \tag{23}$$

High-importance states are transferred from L2 to L3 during idle cycles.

3.6 Critic Model for Hallucination Detection

The critic C_ϕ is a 1B-parameter model that takes the query $\mathbf{q}$, generated response $\mathbf{r}$, and retrieved facts $\mathcal{F}$ as input:

$$P_{ver}, C_{ver} = C_\phi([\mathbf{q}; \mathbf{r}; \mathcal{F}]) \tag{24}$$

where $P_{ver} \in [0, 1]$ is the correctness probability and $C_{ver} \in [0, 1]$ is the confidence score.

Training minimizes the calibrated loss:

$$\mathcal{L}_{critic} = \text{BCE}(P_{ver}, y_{true}) + \lambda \|C_{ver} - y_{true}\|_2^2 \tag{25}$$

with $\lambda = 0.5$ for balanced calibration.

4 Training Methodology

4.1 Stage 1: Pre-training

The base MoE model and memory modules are jointly trained on language modeling:

$$\mathcal{L}_{total} = \mathcal{L}_{LM} + \alpha \mathcal{L}_{bal} \tag{26}$$

We use the AdamW optimizer [26] with learning rate 3×10^{-4}, batch size 512, and mixed-precision training (bf16). Training corpus consists of 2T tokens from diverse sources (web text, books, code).

4.2 Stage 2: Meta-Controller Reinforcement Learning

The Meta-Controller is optimized via Proximal Policy Optimization (PPO) [27] to maximize expected reward:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \gamma^t R_t \right] \tag{27}$$

Reward Function:

$$R_t = w_1 \nVdash_{acc} - w_2 \frac{T_{lat}}{T_{base}} - w_3 \mathcal{C}_{compute} + w_4 \mathcal{Q}_{calib} \tag{28}$$

where:

- $\nVdash_{acc}$: Accuracy (1 if correct, -1 otherwise)
- T_{lat}/T_{base}: Normalized latency (vs. full model inference)
- $\mathcal{C}_{compute} = \frac{\text{active\_params}}{\text{total\_params}}$: Compute cost
- $\mathcal{Q}_{calib} = -(u_t - (1 - y_{true}))^2$: Calibration quality

Weights: $w_1 = 1.0, w_2 = 0.3, w_3 = 0.2, w_4 = 0.5$.

PPO Update: The policy is updated by minimizing the clipped surrogate loss:

$$\mathcal{L}^{PPO}(\theta) = -\mathbb{E}_t \left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{29}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the Generalized Advantage Estimate [28], and $\epsilon = 0.2$ is the clipping parameter.

4.3 Stage 3: Critic Fine-Tuning

The critic is fine-tuned on curated hallucination datasets (HaluEval [29], FaithDial [30]) with:

- Positive examples: Factually correct responses with supporting evidence
- Negative examples: Plausible but incorrect outputs with contradictory evidence

5 Experimental Setup

5.1 Baselines

We compare HMST against:

- **Llama-3-8B** [2]: Dense transformer baseline
- **Mixtral-8x7B** [15]: Sparse MoE without memory augmentation
- **Mamba-2.8B** [11]: Pure SSM architecture
- **Llama-3-8B + RAG**: Dense model with FAISS retrieval

5.2 Datasets

- **LongBench** [31]: Long-context QA (16K-128K tokens)
- **TruthfulQA** [32]: Hallucination-prone questions
- **HaluEval** [29]: Synthetic hallucination detection
- **NarrativeQA** [33]: Book-length comprehension

5.3 Metrics

1. **Accuracy:** Exact match / F1 score on QA tasks
2. **Hallucination Rate:** Fraction of factually incorrect generations
3. **Latency:** Wall-clock inference time (A100 GPU)
4. **FLOPs:** Floating-point operations per token
5. **Calibration Error:** Expected Calibration Error (ECE) [34]

6 Results

6.1 Long-Context Performance

Table 1 shows results on LongBench. HMST achieves competitive accuracy while reducing compute by 58% relative to dense baselines.

Table 1: Performance on LongBench (16K context)

Model	Accuracy (%)	Latency (s)	FLOPs (T)
Llama-3-8B	68.3	2.45	12.4
Mixtral-8x7B	71.2	1.89	8.7
Mamba-2.8B	63.8	0.52	3.1
Llama-3 + RAG	69.7	2.91	13.2
HMST-12B	**72.8**	**1.06**	**5.2**

6.2 Hallucination Mitigation

On TruthfulQA, HMST reduces hallucination rate by 15% compared to Llama-3 baseline (Table 2).

Table 2: TruthfulQA Results

Model	Truthful (%)	Informative (%)
Llama-3-8B	42.1	58.3
Mixtral-8x7B	46.8	61.2
Llama-3 + RAG	49.3	62.7
HMST-12B	**57.2**	**65.4**
HMST + Critic	**61.5**	**64.1**

Table 3: Ablation Study on LongBench

Configuration	Accuracy (%)	Latency (s)
HMST (Full)	72.8	1.06
- L3 Semantic	69.4	0.98
- L2 Episodic	68.1	1.12
- Meta-Controller	65.7	1.34
Base MoE only	63.2	0.87

6.3 Ablation Studies

Table 3 demonstrates the contribution of each memory tier.

6.4 Scaling Analysis

Figure **??** shows that HMST maintains sub-quadratic complexity as context length increases, while dense baselines exhibit quadratic growth.

7 Discussion

7.1 Computational Efficiency

The hierarchical memory design enables graceful degradation: simple queries bypass expensive retrieval (Gate 1), while complex reasoning activates full memory hierarchy. RL optimization learns query-specific routing policies, achieving $2.3\times$ speedup on average.

7.2 Hallucination Mechanisms

The critic model demonstrates strong calibration (ECE = 0.08), effectively filtering 73% of hallucinated outputs when verification gate $g_{ver} > 0.7$. Error analysis reveals remaining failures occur primarily in:

1. Multi-hop reasoning requiring transitive fact composition
2. Temporal reasoning over evolving entity states
3. Commonsense inference beyond retrieved evidence

7.3 Limitations

- **Training Cost:** RL optimization requires 50K episodes ($\sim$200 A100 GPU-hours).
- **Memory Maintenance:** L3 index rebuilding scales as $O(M \log M)$ for M entries.
- **Cold Start:** System requires warmup period to populate episodic memory.

8 Conclusion

We introduced HMST, a hierarchical memory-augmented transformer that addresses fundamental trade-offs in long-context LLM design. By combining sparse MoE computation with multi-tier memory under RL-based meta-control, HMST achieves superior efficiency and factual accuracy compared to existing approaches. The architecture demonstrates that learned routing policies can dynamically balance compute allocation based on query complexity, opening new research directions in adaptive neural architectures.

Future work will explore: (1) scaling memory capacity to 10M+ entries, (2) continual learning protocols for memory consolidation, and (3) integration with tool-augmented reasoning systems.

Broader Impact

HMST's hallucination mitigation capabilities could improve reliability in high-stakes applications (medical QA, legal reasoning). However, the critic model may introduce brittleness if trained on biased datasets, potentially amplifying systemic biases. Deployment should include human-in-the-loop verification for critical decisions.

References

[1] T. Brown et al., "Language Models are Few-Shot Learners," *NeurIPS*, 2020.

[2] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv:2302.13971*, 2023.

[3] Anthropic, "Claude 3 Model Family," Technical Report, 2024.

[4] A. Vaswani et al., "Attention is All You Need," *NeurIPS*, 2017.

[5] T. Dao et al., "FlashAttention: Fast and Memory-Efficient Exact Attention," *NeurIPS*, 2022.

[6] N. Liu et al., "Lost in the Middle: How Language Models Use Long Contexts," *TACL*, 2023.

[7] Y. Zhang et al., "Siren's Song in the AI Ocean," *EMNLP*, 2023.

[8] Z. Ji et al., "Survey of Hallucination in Natural Language Generation," *ACM Comp. Surveys*, 2023.

[9] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP," *NeurIPS*, 2020.

[10] A. Gu et al., "Efficiently Modeling Long Sequences with Structured State Spaces," *ICLR*, 2022.

[11] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," *arXiv:2312.00752*, 2023.

[12] R. Child et al., "Generating Long Sequences with Sparse Transformers," *arXiv:1904.10509*, 2019.

[13] I. Beltagy et al., "Longformer: The Long-Document Transformer," *arXiv:2004.05150*, 2020.

[14] W. Fedus et al., "Switch Transformers: Scaling to Trillion Parameter Models," *JMLR*, 2022.

[15] A. Q. Jiang et al., "Mixtral of Experts," *arXiv:2401.04088*, 2024.

[16] O. Lieber et al., "Jamba: A Hybrid Transformer-Mamba Language Model," *arXiv:2403.19887*, 2024.

[17] A. Graves et al., "Neural Turing Machines," *arXiv:1410.5401*, 2014.

[18] J. Weston et al., "Memory Networks," *ICLR*, 2015.

[19] G. Izacard et al., "Atlas: Few-shot Learning with Retrieval Augmented Language Models," *arXiv:2208.03299*, 2022.

[20] P. Manakul et al., "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection," *EMNLP*, 2023.

[21] L. Gao et al., "RARR: Researching and Revising What Language Models Say," *ACL*, 2023.

[22] L. Kuhn et al., "Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation," *ICLR*, 2023.

[23] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv:1606.08415*, 2016.

[24] E. Jang et al., "Categorical Reparameterization with Gumbel-Softmax," *ICLR*, 2017.

[25] J. Johnson et al., "Billion-scale Similarity Search with GPUs," *IEEE Trans. Big Data*, 2019.

[26] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” *ICLR*, 2019.

[27] J. Schulman et al., “Proximal Policy Optimization Algorithms,” *arXiv:1707.06347*, 2017.

[28] J. Schulman et al., “High-Dimensional Continuous Control Using Generalized Advantage Estimation,” *ICLR*, 2016.

[29] Y. Li et al., “HaluEval: A Large-Scale Hallucination Evaluation Benchmark,” *EMNLP*, 2023.

[30] N. Dziri et al., “FaithDial: A Faithful Benchmark for Information-Seeking Dialogue,” *TACL*, 2022.

[31] Y. Bai et al., “LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding,” *arXiv:2308.14508*, 2023.

[32] S. Lin et al., “TruthfulQA: Measuring How Models Mimic Human Falsehoods,” *ACL*, 2022.

[33] T. Kociský et al., “The NarrativeQA Reading Comprehension Challenge,” *TACL*, 2018.

[34] C. Guo et al., “On Calibration of Modern Neural Networks,” *ICML*, 2017.