

EXPLICITLM: DECOUPLING KNOWLEDGE FROM PARAMETERS VIA EXPLICIT MEMORY BANKS

Anonymous authors

Paper under double-blind review

ABSTRACT

Large language models (LLMs) universally suffer from knowledge staleness and lack of interpretability due to their implicit knowledge storage paradigm, where information is distributed across network parameters in an entangled, non-addressable manner. This fundamental limitation prevents targeted knowledge updates, verification of stored information, and understanding of model reasoning processes. We propose ExplicitLM, a novel architecture that fundamentally reimagines knowledge storage in language models through an explicit, interpretable memory bank system. Our key innovation introduces a million-scale external memory bank where each entry stores human-readable knowledge as token sequences, enabling direct inspection and modification of the model’s knowledge base. To efficiently access this massive repository, we design a differentiable two-stage retrieval mechanism that enables end-to-end training while maintaining discrete knowledge selection, combining efficient coarse-grained filtering with product key decomposition (reducing computational complexity from $\mathcal{O}(N \cdot |I|)$ to $\mathcal{O}(\sqrt{N} \cdot |I|)$) and fine-grained similarity matching through Gumbel-Softmax. Drawing inspiration from dual-system cognitive theory, we partition knowledge into frozen explicit facts (20%) and learnable implicit patterns (80%), maintained through an Exponential Moving Average update strategy that ensures training stability. Extensive experiments demonstrate that ExplicitLM achieves up to 43.67% improvement in knowledge-intensive tasks compared to standard Transformers, with particularly pronounced gains in low-data regimes ($3.62\times$ improvement with 10k samples). Our analysis reveals strong correlations between memory retrieval success and task performance, with correctly predicted samples achieving 49% higher memory hit rates. Unlike traditional RAG systems with frozen retrieval components, our jointly optimized architecture demonstrates that interpretable, updatable language models can maintain competitive performance while providing unprecedented transparency into their knowledge utilization.

1 INTRODUCTION

Contemporary large language models (LLMs) universally suffer from knowledge staleness, with internally stored knowledge frozen at training completion Cheng et al. (2024); Singh et al. (2025). This temporal limitation creates a widening gap between static model knowledge and dynamic real-world information. Consider the U.S. presidency: Joe Biden served until January 2025, when Donald Trump assumed office. Models trained before this transition perpetually provide outdated answers, unable to reflect real-time changes. Post-training, this knowledge ossification accumulates across countless facts—political leadership, scientific discoveries, economic indicators, and technological standards—severely undermining model reliability in practical applications Mousavi et al. (2024). Knowledge updating thus emerges as critical: models require mechanisms to incorporate temporal factual changes to maintain utility and trustworthiness in real-world deployments.

Current approaches to acquiring or updating external knowledge primarily rely on two paradigms: real-time querying through Model Context Protocol (MCP) tools Hou et al. (2025), or knowledge augmentation via Retrieval-Augmented Generation (RAG) techniques Lewis et al. (2020). However, MCP-based methods exhibit several critical limitations. First, real-time querying introduces substantial inference latency, degrading user experience Singh et al. (2025). Second, dependency on

external APIs compromises system robustness Li et al. (2025). RAG techniques, though partially mitigating knowledge updating challenges, face persistent obstacles: the relevance between retrieved documents and queries remains difficult to ensure, the inherent misalignment between retrieval and generation objectives yields suboptimal performance, and the maintenance and updating of external knowledge bases incurs substantial engineering overhead Salemi & Zamani (2024). These limitations collectively motivate the need for more efficient and integrated approaches to knowledge acquisition and updating in language models.

The fundamental barrier to direct manipulation of model-internal knowledge stems from the implicit knowledge storage paradigm in current language models. Research demonstrates that LLM knowledge is predominantly distributed across Feed-Forward Network (FFN) layers of the Transformer architecture Geva et al. (2021); Meng et al. (2022); Dai et al. (2022). Unlike traditional databases with discrete, addressable locations, each piece of LLM knowledge emerges from collective parameter interactions across all FFN layers, creating highly entangled representations that cannot be independently isolated or modified. This transforms knowledge update into a formidable challenge: modifying a single fact theoretically requires recalibrating weights throughout the entire network—a practically infeasible task risking catastrophic interference with other stored knowledge. This “black-box” nature prevents both verification of acquired knowledge and targeted correction of problematic content. During pre-training on massive corpora, models inevitably absorb misinformation, outdated content, or harmful material Perełkiewicz & Poświata (2024), yet inability to precisely locate and excise such knowledge allows errors to persist and propagate through outputs, fundamentally undermining reliability and trustworthiness.

More critically, implicit knowledge storage fundamentally impedes interpretability. When generating predictions, researchers cannot trace specific knowledge foundations underlying model reasoning. We cannot determine which facts inform reasoning nor verify reasoning step correctness. This opacity constrains understanding of model behavior and poses fundamental challenges to building trustworthy, interpretable AI systems. In high-reliability domains like medical diagnosis Ennab & Mcheick (2024) and legal consultation Latif (2025), this interpretability lack becomes a primary deployment barrier.

Motivated by these observations, we propose a novel language model architecture incorporating an explicit memory bank. The core innovation lies in transforming traditional implicit knowledge storage into an explicit, interpretable knowledge management system. By introducing accessible Memory Bank layers at each model layer, we enable dynamic retrieval and utilization of external knowledge while, more importantly, achieving transparent knowledge management. Our main contributions are summarized as follows:

- We propose an explicit knowledge storage architecture based on Memory Banks, enabling each knowledge entry in the model’s repository to be decoded into human-readable text format, fundamentally addressing the interpretability limitations of traditional models.
- We design a differentiable two-stage retrieval mechanism that combines discrete knowledge selection with continuous gradient flow, enabling end-to-end training of the memory-augmented architecture while maintaining retrievable interpretability and low computational cost.
- We propose ExplicitLM, a novel architecture that enables explicit retrieval and interpretation of model knowledge while achieving superior answer accuracy compared to standard Transformer baselines.

2 RELATED WORK

2.1 LLM ARCHITECTURE DEVELOPMENT

The evolution of large language model architectures began with BERT Devlin et al. (2019), which introduced bidirectional pre-training through masked language modeling, while GPT-2 Radford et al. demonstrated the power of scaling autoregressive transformers. T5 Raffel et al. (2020) unified various NLP tasks into a text-to-text framework, and GPT-3 Brown et al. (2020) showed emergent few-shot learning capabilities at 175B parameters. Subsequent developments include PaLM Chowdhery et al. (2023) scaling to 540B parameters with improved training efficiency, LLaMA Touvron et al.

(2023) achieving strong performance with smaller models through careful data curation, and GPT-4 Achiam et al. (2023) advancing multimodal capabilities. Recent architectural innovations have explored alternatives to standard transformers: RWKV Peng et al. (2023) combines RNN efficiency with transformer-level performance through linear attention mechanisms, Mamba Gu & Dao (2023) leverages selective state space models for efficient long-context modeling with linear complexity, while Mixtral Jiang et al. (2024) employs sparse mixture-of-experts for improved parameter efficiency.

2.2 KNOWLEDGE EDITING AND UPDATING

Knowledge editing in large language models to eliminate errors remains an emerging research area, with existing approaches divided into parameter-efficient and parameter-augmented methods. Parameter-efficient approaches focus on updating knowledge without additional parameters: Li et al. (2023) introduces KAFT (Knowledge-Augmented Fine-Tuning), a data augmentation strategy incorporating diverse contexts (relevant, irrelevant, and counterfactual) for fine-tuning to reduce knowledge errors, while Onoe et al. (2023) constructs datasets to evaluate whether different methods can successfully inject specific facts and enable reasoning based on them. Parameter-augmented methods introduce additional components: Dong et al. (2022) employs CaliNet, training key-value calibration memory slots with similar architecture to FFN but smaller intermediate dimensions; Wang et al. (2024) embeds memory pools containing compressed knowledge tokens at each layer with update functions, though lacking interpretability; Mitchell et al. (2022) prepends a knowledge classifier to existing models, routing queries to either an explicitly stored and updatable database with a specialized model or the standard LLM, achieving explicit storage but sacrificing end-to-end neural architecture coherence.

3 MEMORY BANK

3.1 MEMORY THEORY

Drawing from dual-system cognitive theory Gowda et al. (2025), we partition language model knowledge into two distinct yet complementary phases analogous to human procedural-declarative memory dichotomy.

Implicit Knowledge: This encompasses linguistic grammar rules, syntactic structures, and semantic associations that resist explicit formalization. Examples include nuanced aspects of human expression patterns and implicit connections between complex concepts that emerge from cultural and contextual understanding. Such knowledge exhibits high abstraction and ambiguity, necessitating statistical learning from large-scale data.

Explicit Knowledge: This comprises factual knowledge, entity relationships, and time-sensitive information amenable to explicit representation. Examples include "The President of the United States is Trump" (not Biden) and "The Eiffel Tower stands 324 meters tall." Such knowledge possesses clear truth conditions and update requirements, making it suitable for storage in editable memory banks.

This dual-system design offers distinct advantages: implicit knowledge, acquired through deep learning, ensures robust language understanding and generation capabilities; explicit knowledge, through structured storage, enables interpretability and updatability. The synergistic integration of both systems enables models to maintain powerful linguistic capabilities while flexibly managing and updating factual knowledge.

3.2 STORAGE ARCHITECTURE

Let $\mathcal{M} \subseteq \mathbb{Z}^{1 \times L}$, $|\mathcal{M}| = N$ denote our Memory Bank tensor, where $N = 10^6$ represents the knowledge capacity and $L = 16$ denotes the maximum sequence length. Each entry $\mathbf{m}_i \in \mathcal{M}$ stores a discrete knowledge unit as token indices, with elements $m_{ij} \in \mathcal{V}$, where \mathcal{V} is codebook.

We employ a tokenizer-based bidirectional mapping scheme. The encoding function $\text{Tokenize} : \mathcal{S} \rightarrow \mathbb{Z}^{1 \times L}$ converts knowledge strings $s \in \mathcal{S}$ to token indices for storage: $\mathbf{m}_i = \text{Tokenize}(s_i) = [t_1^{(i)}, t_2^{(i)}, \dots, t_L^{(i)}]$ where $t_j^{(i)} \in \mathcal{V}$. During retrieval, the embedding function $\text{Embed} : \mathbb{Z}^{1 \times L} \rightarrow$

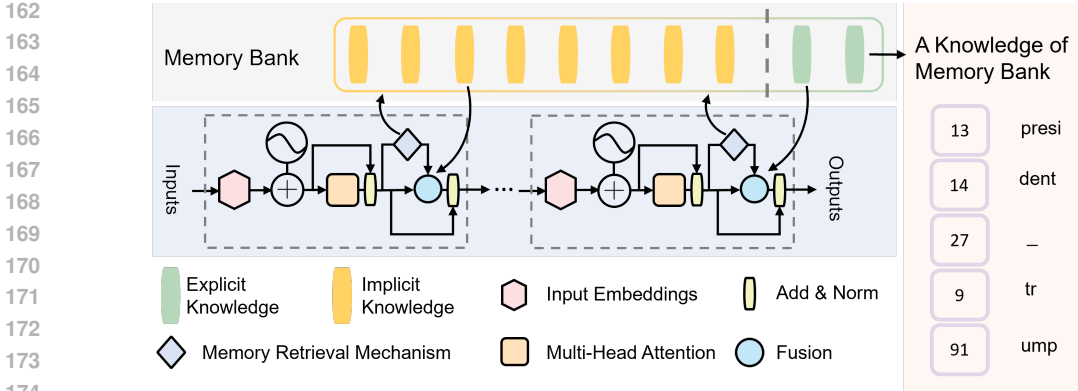


Figure 1: Overall architecture of ExplicitLM. The blue region shows the multi-layer transformer blocks. The gray region represents the shared Memory Bank accessed by all layers, where each layer can retrieve knowledge via the Memory Retrieval Mechanism (Section 3.4) from Explicit Knowledge (green) or Implicit Knowledge (yellow) partitions. The orange region shows a sample knowledge entry from the Memory Bank—a sequence of token indices of length L directly convertible to human-readable text.

$\mathbb{R}^{d \times L}$ transforms stored indices back to semantic representations: $\mathbf{E}_i = \text{Embed}(\mathbf{m}_i) = [\mathbf{e}_{t_1^{(i)}}, \mathbf{e}_{t_2^{(i)}}, \dots, \mathbf{e}_{t_L^{(i)}}]$, where $\mathbf{e}_{t_j^{(i)}} \in \mathbb{R}^d$.

3.3 KNOWLEDGE ALLOCATION STRATEGY

Given the memory constraint $|\mathcal{M}| = N$, our approach maintains a fixed-capacity knowledge repository throughout the model’s lifecycle. This design choice ensures predictable memory consumption and eliminates the computational overhead associated with dynamic memory allocation. To effectively utilize this fixed capacity while preserving essential linguistic knowledge, we introduce a partitioning scheme that divides the memory bank into two disjoint subsets: $\mathcal{M} = \mathcal{M}_f \cup \mathcal{M}_u$ where $\mathcal{M}_f \cap \mathcal{M}_u = \emptyset$. The partition is controlled by a freeze rate parameter $\rho \in [0, 1]$, which determines the proportion of memory allocated to each subset.

The frozen knowledge subset \mathcal{M}_f with cardinality $|\mathcal{M}_f| = \rho N$ (we empirically set $\rho = 0.2$ as default) is designated for storing explicit knowledge that can be precisely formulated and verified. During initialization, this subset is populated with curated factual information such as entity relationships, geographical facts, and time-sensitive data that require accurate representation. The explicit nature of this knowledge allows for direct injection of verified information into the memory bank, ensuring factual accuracy from the outset. These entries remain immutable during training to preserve the integrity of the pre-verified knowledge base. Conversely, the updatable knowledge subset \mathcal{M}_u with cardinality $|\mathcal{M}_u| = (1 - \rho)N$ is allocated for implicit knowledge that the model must discover through training. This subset captures linguistic regularities, syntactic patterns, and semantic associations that emerge from statistical learning over large-scale text corpora. The model autonomously determines which grammatical structures and language patterns warrant storage in this dynamic portion of the memory bank. The in-place substitution mechanism maintains the invariant $|\mathcal{M}^{(t)}| = N$ for all time steps t , as updates neither insert new entries nor delete existing ones, thereby preserving constant memory footprint and eliminating the complexity associated with dynamic memory management operations.

To address the gradient discontinuity issue that arises from direct overwriting of knowledge entries in \mathcal{M}_u , we adopt the Exponential Moving Average (EMA) technique from Vector Quantized Variational Autoencoders (VQ-VAE) Van Den Oord et al. (2017), originally developed for codebook updates. Rather than performing abrupt replacements, the EMA mechanism enables progressive updates that maintain training stability. Specifically, for each knowledge entry $\mathbf{m}_i \in \mathcal{M}_u$, we maintain dynamic statistics that allow smooth transitions between old and new knowledge representations. The update rule assigns higher weights to newer information while preserving continuity with existing knowledge, enabling the memory bank to adapt to evolving encoder outputs without

introducing disruptive oscillations. This approach effectively circumvents the non-differentiability inherent in discrete quantization operations, while simultaneously improving both the utilization rate of knowledge entries and the overall reconstruction quality of the stored information.

3.4 MEMORY RETRIEVAL MECHANISM

We propose a hierarchical two-stage retrieval strategy for efficient access to million-scale entries.

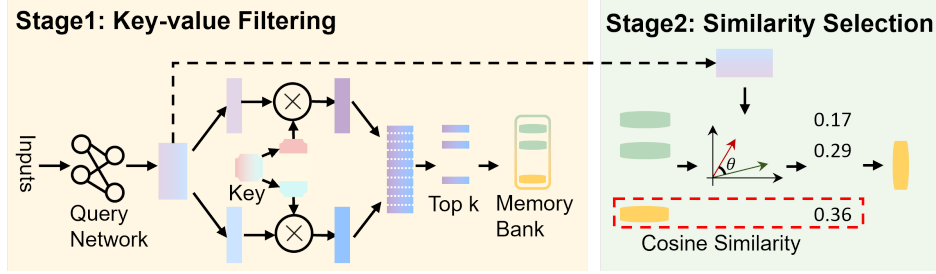


Figure 2: ExplicitLM architecture with memory retrieval mechanism. In Stage 1, both query and key vectors are partitioned along the embedding dimension into two components for efficient retrieval. In Stage 2, cosine similarity is computed between the query and candidate knowledge entries, with the highest-scoring entry selected for retrieval.

Stage 1: Key-value Filtering. Following Million ExpertsHe (2024), we assign product keys $\mathbf{K} := \{k_i\}_{i=1}^N \subset \mathbb{R}^d$ to knowledge entries, with query network q mapping input x to query $q(x)$. This stage generates a candidate set I by retrieving the most relevant entries based on query-key similarities: $I = \text{top-}I\text{-indices}(\{q(x)^\top k \mid k \in \mathbf{K}\})$, where $\text{top-}I\text{-indices}$ denotes the operator that selects the indices of the top- I elements from \mathbf{K} , yielding a candidate set with cardinality $|I|$. To address computational complexity at $N \geq 10^6$, we decompose keys using Cartesian products: $\mathbf{K} = \{[c; c'] \mid c \in \mathbf{C}, c' \in \mathbf{C}'\}$ where $\mathbf{C}, \mathbf{C}' \subset \mathbb{R}^{d/2}$ with $|\mathbf{C}| = |\mathbf{C}'| = \sqrt{N}$, reducing complexity from $\mathcal{O}(N \cdot |I|)$ to $\mathcal{O}(\sqrt{N} \cdot |I|)$, where $|I| \ll \sqrt{N}$.

Stage 2: Similarity Selection. For candidates $i \in I$, we compute cosine similarities $cs_i = \cos(q(x), k_i)$ and apply Gumbel-Softmax for differentiable selection:

$$p_i = \frac{\exp((cs_i + g_i) / \tau)}{\sum_{j \in I} \exp((cs_j + g_j) / \tau)} \quad (1)$$

where $g_i = -\log(-\log(\epsilon_i))$ with $\epsilon_i \sim \text{Uniform}(0, 1)$ and temperature τ . The straight-through estimator enables gradient flow: forward pass selects $\mathbf{m}_{\text{selected}} = \mathbf{m}_{\hat{i}}$ where $\hat{i} = \arg \max_i p_i$, while backward pass uses soft weights $\frac{\partial \mathcal{L}}{\partial q(x)} = \sum_{i \in I} p_i \frac{\partial \mathcal{L}}{\partial \mathbf{m}_i}$, maintaining discrete selection while ensuring end-to-end differentiability for retrieved knowledge $\mathbf{m}_{\text{selected}} \in \mathcal{M}$.

Unlike traditional RAG systems that rely on frozen retrieval components, our mechanism enables joint optimization of retrieval and generation through differentiable selection, allowing the model to learn task-specific retrieval patterns during training.

3.5 JOINT OPTIMIZATION OBJECTIVE

We design a multi-task learning framework that jointly optimizes three complementary losses to balance language modeling capability with effective memory retrieval.

Language Modeling Loss. Following standard practice, we minimize cross-entropy between predicted and ground-truth distributions. For sequence $\mathbf{x} = (x_1, \dots, x_T)$ with vocabulary \mathcal{V} of size V :

$$\mathcal{L}_{\text{CE}} = -\frac{1}{T} \sum_{t=1}^T \log p(x_t | x_{<t}, \mathcal{M}) \quad (2)$$

where $p(x_t|x_{<t}, \mathcal{M})$ denotes the model’s predicted probability conditioned on context $x_{<t}$ and retrieved memories from \mathcal{M} .

Memory Relevance Loss. To ensure semantic alignment between queries and retrieved memories, we maximize weighted cosine similarities. Given query $q(x) \in \mathbb{R}^d$ and retrieved candidates $\{\mathbf{E}_i\}_{i=1}^{|I|}$ with selection weights $\{p_i\}_{i=1}^{|I|}$ from Gumbel-Softmax:

$$\mathcal{L}_{\text{sim}} = -\mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[\sum_{i=1}^{|I|} p_i \cdot \frac{q(\mathbf{x})^T \mathbf{E}_i}{\|q(\mathbf{x})\|_2 \|\mathbf{E}_i\|_2} \right] \quad (3)$$

This loss guides the model toward selecting contextually relevant memories by reinforcing high-similarity retrievals.

Memory Diversity Loss. To prevent retrieval collapse into local regions and expand semantic coverage, we minimize pairwise similarities among the candidates. Let $\hat{\mathbf{E}}_i = \mathbf{E}_i / \|\mathbf{E}_i\|_2$ denote normalized embeddings:

$$\mathcal{L}_{\text{div}} = \frac{2}{|I|(|I|-1)} \sum_{i=1}^{|I|} \sum_{j=1, j \neq i}^{|I|} cs(\hat{\mathbf{E}}_i, \hat{\mathbf{E}}_j) \quad (4)$$

This regularization encourages exploration across diverse memory regions, preventing locally optimal retrieval patterns while maintaining relevance through balanced optimization.

The final objective combines all losses: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{sim}} \mathcal{L}_{\text{sim}} + \lambda_{\text{div}} \mathcal{L}_{\text{div}}$. This joint optimization ensures: (1) accurate next-token prediction through \mathcal{L}_{CE} , (2) semantically coherent retrieval via \mathcal{L}_{sim} , and (3) diverse memory exploration through \mathcal{L}_{div} , yielding an end-to-end trainable knowledge-augmented architecture where memory retrieval and language modeling are deeply integrated.

4 EXPERIMENTS

4.1 DATASET CONSTRUCTION

We construct a 10M-entry multi-source pretraining corpus with strategic sampling ratios optimized for knowledge diversity: **Wikipedia**: Structured encyclopedic knowledge annotated with entity triplets for explicit knowledge graph extraction. These entries form the exclusive source for Memory Bank initialization $\mathcal{M} \subseteq \mathbb{Z}^{1 \times L}$, selected based on knowledge density metrics and factual reliability scores. **Project Gutenberg**: Literary and historical texts providing formal language patterns and narrative structures spanning multiple centuries. **OpenWebText**: Contemporary web text capturing modern linguistic phenomena and informal discourse patterns.

Each entry maintains a unique identifier for provenance tracking. The Memory Bank entries \mathbf{m}_i are mapped to source UUIDs, enabling systematic knowledge updates and verification. Selection criteria prioritize: (i) token-level information density, (ii) factual accuracy via cross-reference validation, and (iii) domain coverage measured by entity distribution.

4.2 EVALUATION TASK DESIGN

We design three complementary tasks to evaluate knowledge utilization from Memory Bank \mathcal{M} : **(i) Object Prediction**: Given subject-predicate pairs from knowledge entries $\mathbf{m}_i \in \mathcal{M}$, predict correct object tokens t_{ji} from candidate set. Accuracy measures entity relationship understanding with 5 distractors in \mathbb{R}^d space. **(ii) Relation Reasoning**: Given entity token pairs (t_{ji}, t_{ki}) from \mathbf{m}_i , infer their semantic relationship. This probes compositional reasoning over stored knowledge structures in \mathcal{M} . **(iii) Fact Verification**: Binary classification of statements derived from memory bank domain. Negative samples generated via token substitution at indices $m_{i,j}$ maintain 50 : 50 class balance. Data partitioning leverages freeze partition: test samples derive exclusively from frozen entries \mathcal{M}_f where $|\mathcal{M}_f| = \rho N$, while training excludes all tokens from $\mathbf{m}_i \in \mathcal{M}_f$. This strict disjoint constraint between \mathcal{M}_f and training data prevents memorization-based evaluation inflation.

4.3 COMPARISON OF DIFFERENT DATA VOLUMES

To systematically evaluate the efficacy of our memory-augmented architecture, we conduct controlled experiments across varying supervised fine-tuning (SFT) data volumes. Both our model and the baseline Transformer undergo identical optimization procedures, with performance assessed on the three tasks defined in Section 4.2. The baseline represents a standard Transformer architecture without memory augmentation, enabling direct attribution of performance gains to our proposed Memory Bank mechanism.

Table 1: Performance comparison between baseline Transformer and our memory-augmented model across different SFT data volumes. Results show accuracy (%) on three knowledge-intensive tasks.

Data Volume	Model	Object Prediction	Relation Reasoning	Fact Verification
10k	Baseline	7.86%	38.27%	61.71%
	Ours	28.42% \uparrow 20.56%	70.02% \uparrow 31.75%	66.03% \uparrow 4.32%
25k	Baseline	22.16%	79.99%	71.49%
	Ours	63.12% \uparrow 40.96%	87.85% \uparrow 7.86%	79.79% \uparrow 8.33%
50k	Baseline	30.23%	83.80%	83.34%
	Ours	73.90% \uparrow 43.67%	90.41% \uparrow 6.61%	86.25% \uparrow 2.91%
75k	Baseline	40.64%	87.66%	86.40%
	Ours	79.76% \uparrow 39.12%	92.12% \uparrow 4.46%	88.74% \uparrow 2.34%
100k	Baseline	56.80%	91.91%	88.92%
	Ours	80.94% \uparrow 24.14%	92.73% \uparrow 0.82%	89.75% \uparrow 0.83%

The experimental results reveal pronounced performance advantages in low-data regimes. At 10k training samples, our model achieves 3.62 \times improvement in Object Prediction and 1.83 \times improvement in Relation Reasoning compared to the baseline. This substantial gap demonstrates that explicit memory retrieval from \mathcal{M} effectively compensates for limited training exposure, particularly for tasks requiring precise entity-level knowledge recall. The Object Prediction task, which directly queries stored triplets from memory entries m_i , exhibits the most consistent improvements across all data scales (24.14% at 100k samples), validating our retrieval mechanism’s effectiveness in accessing specific tokens t_{ji} from the Memory Bank.

4.4 MEMORY BANK HIT RATE ANALYSIS

To empirically validate the effectiveness of our Memory Bank retrieval mechanism, we conduct a fine-grained analysis of layer-wise memory access patterns. Using models trained with varying data volumes from Section 1, we examine the correlation between successful memory retrieval and task performance on Relation Reasoning. For each forward pass, we track whether the retrieval mechanism successfully matches relevant entries from \mathcal{M} at each transformer layer, providing insights into how different layers utilize external memory.

The aggregate hit rates reveal a strong correlation between memory access success and prediction accuracy. Models trained on 100k, 50k, 25k, 10k, and 5k samples achieve overall hit rates of 71%, 65%, 66%, 71%, and 71% respectively for correctly answered samples, where a sample is considered to have "hit" if at least one layer successfully retrieves relevant memory. In stark contrast, incorrectly answered samples exhibit substantially lower hit rates of 23%, 21%, 21%, 22%, and 37% respectively. This 3 \times differential in hit rates between correct and incorrect predictions empirically confirms that successful memory retrieval directly contributes to task performance.

Figure 3 presents the layer-wise decomposition of hit rates, revealing distinct retrieval patterns across the network depth. Both correct and incorrect samples exhibit elevated hit rates at layers L1 and L3, suggesting these layers serve as critical junctures for knowledge integration. The consistency of this pattern across different training data volumes indicates an emergent specialization in the network architecture, where specific layers develop stronger affinity for external memory access.

4.5 IMPACT OF FREEZE RATE ON PERFORMANCE

To investigate freeze rate parameter ρ effects on model performance, we conduct systematic experiments varying ρ while maintaining other hyperparameters constant. The freeze rate controls

378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393

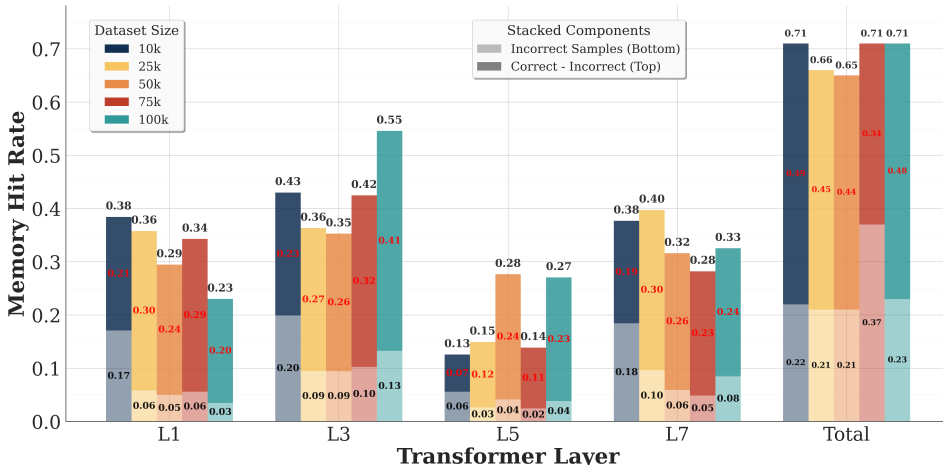


Figure 3: Layer-wise memory hit rates for Relation Reasoning across varying training volumes. Semi-transparent regions indicate hit rates for correctly predicted samples, while opaque regions show hit rates for incorrect predictions. Red annotations display the hit rate differential between correct and incorrect predictions at each layer.

394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418

partition between frozen knowledge \mathcal{M}_f and updatable knowledge \mathcal{M}_u , with $|\mathcal{M}_f| = \rho N$ and $|\mathcal{M}_u| = (1 - \rho)N$. We evaluate performance on Relation Reasoning across different training data volumes to understand how explicit-implicit knowledge balance affects learning dynamics.

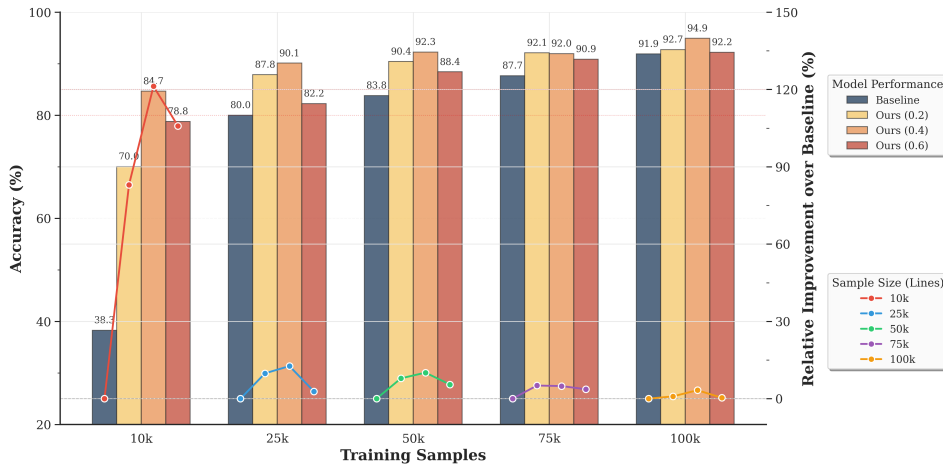


Figure 4: Performance comparison across different freeze rates. The bar chart shows accuracy values under various experimental conditions with different training set sizes. The line plot indicates the relative performance improvement (in percentage) of our method compared to the baseline at different freeze rates for each training set size.

424
425
426
427
428
429
430
431

Figure 4 demonstrates that our memory-augmented architecture consistently outperforms the baseline across all freeze rate configurations. Most pronounced improvements emerge in low-data regimes: with 10k training samples, our method achieves minimum 83% improvement regardless of ρ , highlighting Memory Bank mechanism robustness to hyperparameter selection. Even at 100k samples where baseline reaches 91.91% accuracy, our approach maintains 0.3%-3.3% improvements, confirming explicit memory benefits persist when parametric learning approaches saturation.

The freeze rate-performance relationship exhibits non-monotonic patterns, with optimal performance at $\rho = 0.4$ across most training set sizes. This peak suggests critical balance between explicit knowledge preservation in \mathcal{M}_f and implicit knowledge adaptation in \mathcal{M}_u . Lower freeze

rates ($\rho < 0.4$) potentially compromise core linguistic knowledge stability, allowing excessive updates corrupting fundamental representations. Higher freeze rates ($\rho > 0.4$) restrict model capacity to incorporate task-specific patterns through gradient-based learning, limiting domain-specific adaptation. This trade-off validates our architectural design where frozen entries preserve high-fidelity factual knowledge while updatable entries accommodate evolving linguistic patterns, with optimal partition emerging empirically at approximately 40% frozen knowledge allocation.

4.6 IMPACT OF PERFECT RETRIEVAL ON MODEL PERFORMANCE

To quantify the potential performance gains from improved retrieval accuracy, we conduct controlled experiments comparing autonomous retrieval (Retain) against surgical replacement (Replace) of retrieval results. Based on the critical layers identified in Section 4.5, we intervene at layers L1 and L3 by replacing the top-ranked candidate from the 16 retrieved entries with the oracle knowledge entry most relevant to the correct answer. This experimental design isolates the effect of retrieval quality from other architectural components, providing an upper bound on performance improvements achievable through enhanced retrieval mechanisms.

Table 2: Accuracy comparison between autonomous retrieval (Retain) and surgical replacement of retrieval results at specific layers (Replace) to evaluate the impact of perfect retrieval on model performance.

Data Volume	Model	Object Prediction	Relation Reasoning	Fact Verification
50k	Retain	70.87%	89.87%	85.12%
	Replace	74.49% $\uparrow 3.62\%$	92.12% $\uparrow 2.25\%$	87.24% $\uparrow 2.12\%$
75k	Retain	77.12%	90.25%	88.00%
	Replace	79.85% $\uparrow 2.73\%$	91.87% $\uparrow 1.62\%$	90.25% $\uparrow 2.25\%$
100k	Retain	79.12%	90.50%	90.37%
	Replace	81.00% $\uparrow 1.88\%$	92.25% $\uparrow 1.75\%$	91.12% $\uparrow 0.75\%$

Table 2 demonstrates consistent improvements across all tasks when perfect retrieval is guaranteed, with an average accuracy gain of 2.11 percentage points. The Object Prediction task exhibits the largest improvements (3.62% at 50k samples), consistent with its direct dependence on retrieving specific factual entries from \mathcal{M} . This task directly queries token sequences \mathbf{m}_i for entity relationships, making it most sensitive to retrieval precision. Relation Reasoning shows moderate gains (2.25% at 50k, 1.75% at 100k), suggesting that compositional reasoning benefits from accurate knowledge retrieval but also relies on learned transformations within the network. The diminishing returns observed at larger training volumes (100k samples) indicate that models with more extensive training develop compensatory mechanisms for imperfect retrieval. The average improvement decreases from 2.66% at 50k samples to 1.46% at 100k samples, suggesting that larger training sets enable the model to learn robust representations that partially mitigate retrieval errors.

5 CONCLUSION

We presented ExplicitLM, a novel language model architecture that fundamentally transforms knowledge storage from implicit distributed representations to an explicit, interpretable Memory Bank system. Our approach addresses critical LLM limitations—knowledge staleness, lack of interpretability, and update difficulties—by introducing dual-system design partitioning knowledge into frozen explicit entries and updatable implicit components. Comprehensive experiments demonstrated that ExplicitLM consistently outperforms baseline Transformers across knowledge-intensive tasks, with 20 – 40% improvements in low-data regimes and maintained advantages at scale. Layer-wise hit rate analysis confirmed successful memory retrieval directly correlates with prediction accuracy, validating our two-stage differentiable retrieval mechanism. While current implementation requires manual curation of explicit knowledge entries, this limitation points to promising future directions: developing mechanisms to automatically extract and update explicit knowledge from training data while preserving human readability and interpretability. Such advances would enable models to continuously expand verifiable knowledge bases during training, combining statistical learning benefits with transparent, editable knowledge management—crucial for building trustworthy, maintainable AI systems for real-world deployment.

REFERENCES

- 486
487
488 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale-
489 man, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical
490 report. *arXiv preprint arXiv:2303.08774*, 2023.
- 491 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,
492 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are
493 few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- 494 Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin
495 Van Durme. Dated data: Tracing knowledge cutoffs in large language models. *arXiv preprint*
496 *arXiv:2403.12958*, 2024.
- 497
498 Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam
499 Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm:
500 Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):
501 1–113, 2023.
- 502 Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons
503 in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for*
504 *Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.
- 505 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep
506 bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of*
507 *the North American chapter of the association for computational linguistics: human language*
508 *technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.
- 509
510 Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. Calibrating factual
511 knowledge in pretrained language models. In *Findings of the Association for Computational*
512 *Linguistics: EMNLP 2022*, pp. 5937–5947, 2022.
- 513
514 Mohammad Ennab and Hamid Mcheick. Enhancing interpretability and accuracy of ai models in
515 healthcare: a comprehensive review on challenges and future directions. *Frontiers in Robotics*
516 *and AI*, 11:1444763, 2024.
- 517
518 Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are
519 key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural*
Language Processing, pp. 5484–5495, 2021.
- 520
521 Shruthi Gowda, Bahram Zonooz, and Elahe Arani. Dual cognitive architecture: Incorporating biases
522 and multi-memory systems for lifelong learning. *Transactions on Machine Learning Research*,
523 2025.
- 524
525 Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv*
preprint arXiv:2312.00752, 2023.
- 526
527 Xu Owen He. Mixture of a million experts. *arXiv preprint arXiv:2407.04153*, 2024.
- 528
529 Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. Model context protocol (mcp): Land-
scape, security threats, and future research directions. *arXiv preprint arXiv:2503.23278*, 2025.
- 530
531 Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bam-
532 ford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al.
533 Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.
- 534
535 Youssef Abdel Latif. Hallucinations in large language models and their influence on legal reason-
536 ing: Examining the risks of ai-generated factual inaccuracies in judicial processes. *Journal of*
Computational Intelligence, Machine Reasoning, and Decision-Making, 10(2):10–20, 2025.
- 537
538 Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,
539 Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented gener-
ation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:
9459–9474, 2020.

- 540 Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu,
541 and Sanjiv Kumar. Large language models with controllable working memory. In *Findings of the*
542 *Association for Computational Linguistics: ACL 2023*, pp. 1774–1793, 2023.
- 543 Zhihao Li, Kun Li, Boyang Ma, Minghui Xu, Yue Zhang, and Xiuzhen Cheng. We urgently need
544 privilege management in mcp: A measurement of api usage in mcp ecosystems. *arXiv preprint*
545 *arXiv:2507.06250*, 2025.
- 546 Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual
547 associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.
- 548 Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-
549 based model editing at scale. In *International Conference on Machine Learning*, pp. 15817–
550 15831. PMLR, 2022.
- 551 Seyed Mahed Mousavi, Simone Alghisi, and Giuseppe Riccardi. Dyknow: Dynamically verify-
552 ing time-sensitive factual knowledge in llms. In *Findings of the Association for Computational*
553 *Linguistics: EMNLP 2024*, pp. 8014–8029, 2024.
- 554 Yasumasa Onoe, Michael Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. Can
555 lms learn new entities from descriptions? challenges in propagating injected knowledge. In
556 *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume*
557 *1: Long Papers)*, pp. 5469–5485, 2023.
- 558 Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman,
559 Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for
560 the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.
- 561 Michał Perełkiewicz and Rafał Poświata. A review of the challenges with massive web-mined
562 corpora used in large language models pre-training. In *International Conference on Artificial*
563 *Intelligence and Soft Computing*, pp. 153–163. Springer, 2024.
- 564 Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language
565 models are unsupervised multitask learners.
- 566 Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi
567 Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text
568 transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- 569 Alireza Salemi and Hamed Zamani. Evaluating retrieval quality in retrieval-augmented generation.
570 In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development*
571 *in Information Retrieval*, pp. 2395–2400, 2024.
- 572 Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. A survey of the model context
573 protocol (mcp): Standardizing context to enhance large language models (llms). 2025.
- 574 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée
575 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and
576 efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- 577 Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in*
578 *neural information processing systems*, 30, 2017.
- 579 Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng
580 Li, Xian Li, Bing Yin, et al. Memoryllm: Towards self-updatable large language models. *arXiv*
581 *preprint arXiv:2402.04624*, 2024.

588 A APPENDIX

589 A.1 REPRODUCIBILITY STATEMENT

590 We are committed to ensuring the full reproducibility of our work. All experiments presented in this
591 paper can be reproduced using the code and configurations provided in our anonymous repository
592 (ExplicitLM). All experiments were conducted on NVIDIA A100 GPUs.
593

594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647

A.2 AI ASSISTANCE STATEMENT

We declare that AI-based tools were used solely for language polishing purposes in this work. Specifically, after completing the initial draft entirely through human effort, we employed AI assistance exclusively for grammatical refinement and improving the clarity of English expression to meet academic writing standards. The AI tools did not contribute to: (1) the generation or development of research ideas, including the core concept of ExplicitLM and the memory bank mechanism; (2) the design of experiments or methodology; (3) the analysis or interpretation of results; (4) the drafting of original content or scientific arguments; or (5) any mathematical derivations or technical contributions. All intellectual contributions, from conceptualization to initial manuscript preparation, were performed by the human authors. The use of AI was limited to post-writing language enhancement, similar to traditional proofreading services, ensuring that non-native English speakers can present their research with appropriate linguistic quality while maintaining complete authorship and originality of the scientific content.