ChatGPT

A Temporally Persistent, Memory-Bounded AGI with SAMR and Live Streaming

Abstract: We propose an AGI framework that treats *intelligence as a temporally persistent, memory-bounded process* with a context-defined lifespan. In this view, an LLM becomes a continuous agent that actively manages its limited "working memory" by **Selective Active Memory Recall (SAMR)** – a mechanism for choosing which tokens to retain, forget, or compress in its context. Simultaneously, *live-stream integration* feeds continuous sensory data (audio, video, text) as token streams into the model in real time. Together, SAMR and streaming transform a static prompt–response LLM into a proto-AGI: it maintains a coherent identity over time, gracefully switches tasks, and adapts via dynamic memory management. We compare this approach to Retrieval-Augmented Generation (RAG) [1] [2], MemGPT's OS-inspired memory tiers [3], and other long-context or continual-learning methods, highlighting fundamental differences. A prototype architecture (conceptual diagram below) illustrates how sensors feed a live token stream into the LLM; a SAMR controller and memory module perform read/write on a persistent memory store; and the model incrementally updates its context. We discuss theoretical implications and limitations (e.g. context-window–defined lifespan) and outline extensions such as memory distillation and lifelong weight updates. Our framework emphasizes scientific rigor and cites current literature to support each component.

Introduction

Classical LLM usage is *stateless*: after generating a response, an LLM "forgets" unless prior information is explicitly carried in the next prompt. In practice, an LLM's memory is strictly bounded by its context window [4]. This statelessness limits continuous interaction, long-horizon tasks, and a sense of persistent identity. We redefine "AGI" not as a monolithic model with infinite capacity, but as an **agentic system** with *temporal persistence* and *bounded working memory*. Specifically, our AGI runs continuously, with a finite context size that defines its *lifespan* of immediate awareness. To appear "intelligent" over time, it must manage its own memory: continually compressing or offloading old information while preserving key facts. Recent work emphasizes this need: "LLMs remain fundamentally stateless: their memory is bounded by a finite context window and any information that falls outside this window is forgotten" [4]. Thus, to build persistent agents, we must endow LLMs with explicit memory management.

Two key innovations enable this: **Selective Active Memory Recall (SAMR)** and **live-stream token integration**. SAMR is an LLM-driven mechanism that actively decides which parts of its context to keep, discard, or summarize at each step. Analogously to a human focusing attention or rehearsing important items, SAMR treats the context window like an active cache. Meanwhile, live-stream integration continually converts sensory inputs (speech, video frames, text streams) into tokens fed into the LLM. This makes the LLM a true *streaming agent* rather than a static Q&A module. We show that combining these yields **proto-AGI behavior**: the model develops a self-consistent identity across time, handles errors by revisiting memory, switches tasks without loss of prior intent, and adapts its memory online.

We contrast this framework with existing approaches. Standard RAG pipelines [1] [2] augment a single prompt with retrieved documents, but remain episodic and passive. Systems like *MemGPT* use multi-tier

memory caches (inspired by operating systems) to extend context [3], but still rely on rigid tier management. Long-context transformers or "attention-sink" methods [5] [6] maintain performance over long dialogs without expanding memory. Continual-learning LLM work (e.g. fine-tuning or RL-based memory controllers [7]) focus on weight changes rather than runtime context management. Our system differs fundamentally: it embeds memory control within the inference loop, treating context as a dynamic, adaptive resource.

Finally, we sketch a prototype architecture (Figure 1) showing the **input stream**, **SAMR loop**, **memory manager**, and the live LLM core. We discuss limits (the context window still bounds lifespan) and future extensions (e.g. distilling memory into model weights, modular updates). Our goal is a scientifically grounded proposal for a memory-augmented, streaming LLM agent that approaches general intelligence through persistence and adaptation.

Redefining AGI: Persistent and Context-Bounded

AGI is often described as a system that can perform any intellectual task. We refine this: *An AGI is a temporally extended agent whose cognitive state evolves over time, subject to finite memory constraints.* In effect, the context window becomes the agent's "life at a glance". When more tokens arrive than fit, older context must be offloaded or compressed. This creates a natural **lifespan** for in-context memory: after N tokens (window size), information beyond is "forgotten." As Memory-R1 notes, "any information that falls outside [the context] window is forgotten, preventing [LLMs] from maintaining knowledge across long conversations" [4]. We embrace this: an AGI's continuity is maintained not by infinite memory, but by intelligent memory management.

Under this view, each session is not a fresh state but a continuation. The agent *can* recall old experiences if they were stored, even after they have cycled out of the context. Thus, AGI requires external or compressed memory. Unlike static LLMs or single-turn models, our AGI periodically integrates new experiences into a long-term store and uses SAMR to rehearse them. Its identity emerges from the evolving contents of both its context and persistent memory. In short, we propose AGI = **Agent + Memory + Time**: a looped process where the LLM's context is a moving window of awareness, managed by an agentic controller.

Selective Active Memory Recall (SAMR)

We introduce **Selective Active Memory Recall (SAMR)** as the core mechanism for memory control. SAMR is a learned (or prompted) policy by which the LLM continuously decides how to handle each new token or thought. When new information arrives, SAMR evaluates its relevance: it may *retain* key tokens in the context, *discard* irrelevant ones, or *compress/summarize* chunks into shorter representations (or into the memory manager). We can view SAMR as an active CRUD controller: it chooses *ADD*, *UPDATE*, *DELETE* or *NOOP* for memory entries [7]. For example, if the agent hears a repeated instruction, SAMR might update an existing memory rather than store a duplicate. In essence, SAMR simulates the focusing of attention and rehearsal in working memory (similar to human executive control).

Conceptually, SAMR is an LLM-driven memory module. It can be implemented by prompting the LLM to periodically analyze its context and output memory-management actions. For instance, at regular intervals the model could generate a summary of the earliest k tokens and store it, then purge them from context. Alternatively, a separate "memory manager" LLM (as in Memory-R1 [7] [8]) could decide on memory

operations based on context embeddings. Crucially, SAMR differs from static recall: instead of blindly appending all conversation to memory (as in basic logs [7]), it selectively filters out noise and prioritizes high-signal information.

While SAMR is novel, it builds on recent work. Memory-R1 trains an LLM to perform explicit memory operations (Add, Update, Delete, Noop) with reinforcement learning, achieving more coherent memory maintenance [8] [7]. Our SAMR generalizes this: it can run continuously (not just QA tasks), and can compress multiple tokens at once. In effect, SAMR blurs the line between inference and memory management. Over time, this yields a *self-aware* agent: by deciding what to remember, the agent curates its own identity and knowledge.

Live Stream Token Integration

To realize persistent intelligence, the LLM must process real-time inputs. We propose **live stream integration**: all sensor data (audio, video, or textual feeds) are tokenized on-the-fly and injected into the model as a continuous stream. This treats the LLM like an online interpreter of its environment. Recent multimodal LLMs hint at this capability: for example, *Flow-Omni* processes raw audio into continuous speech tokens, enabling real-time speech-to-speech interaction with low latency [9]. Likewise, vision-language models are advancing at streaming video; *VideoStreaming* encodes long video as a sequence of clips with propagated memory, maintaining a fixed-length representation for arbitrarily long footage [10].

Concretely, a speech stream would be fed through a continuous speech tokenizer (e.g. a model like Whisper or a learned codec) that produces a token every few milliseconds. These tokens enter the context window just like any text. Periodically, SAMR will compress older audio context into higher-level summaries (e.g. salient sentences or detected events), making room for new audio. Similarly, a video feed could be tokenized by an image encoder (e.g. generating caption or object tokens per frame) and streamed into the LLM [10]. In all cases, because tokens are continually streaming, the system can *in principle* run indefinitely, limited only by how well SAMR manages context.

Importantly, live streaming does **not** magically extend the context size. As *StreamingLLM* demonstrates, one can process “infinite” input by sliding and caching only recent tokens plus strategic “attention sinks,” but the LLM never sees all past tokens at once [5] [6]. Our approach similarly never expands the context window; instead, it fuses streaming with memory management. The LLM generates responses or actions based on the latest context plus any memories retrieved. We emphasize: streaming integration means the LLM can be continuously aware of its inputs, but *long-term understanding* still relies on storing and recalling information via SAMR.

SAMR + Streaming: Proto-AGI Behaviors

By combining SAMR with live streams, the model gains several proto-AGI properties. **Self-consistent identity:** as the agent converses or perceives over time, SAMR writes persistent memories about its past actions and history. The agent can recall and build on past experiences, yielding continuity. For example, if the agent “was told” a system prompt at the start, SAMR can maintain that persona information rather than having it expire. The result is a coherent “self” across interactions.

Error resilience: real-time data and an evolving memory allow feedback loops. If the agent makes a mistake, later context or user corrections are added. SAMR can revise previous memory entries (e.g. using *UPDATE* instead of *ADD*) [7], preventing contradictions. If something is forgotten, the agent might request clarification (leveraging streaming input). In short, the agent can recover from errors by consulting its memory and new input, rather than failing silently as a pure stateless LLM would.

Graceful task-switching: humans switch tasks by archiving one task context and loading another. Our system can do similarly. When a new task begins, SAMR can snapshot the old context (e.g. compress it into memory) and shift focus to the new one. Later, it can restore relevant fragments from memory. This avoids confusion when juggling multiple threads of work.

Adaptive memory growth: Over time the memory store grows organically. SAMR might keep richer detail about frequently referenced topics, while pruning trivia. Thus the agent's memory effectively *adapts* to what it finds important. We hypothesize that even without training, such an agent could show emergent learning: repeated exposure to a fact leads SAMR to reinforce it in memory. (Research like A-Mem explicitly evolves memory graphs with new experiences [11] [12]; SAMR would serve a related purpose of dynamic knowledge evolution.)

These behaviors make the agent far more robust than a static chatbot. They mirror how cognitive architectures treat memory: attention and rehearsal determine retention, and continuous perception informs action [10] [9]. While true AGI remains elusive, our framework can demonstrate "lifelong" consistency and adaptability in restricted domains (e.g. day-long conversations, multi-session tasks) by design.

Relation to Prior Approaches

- **Retrieval-Augmented Generation (RAG):** RAG pipelines enhance LLM answers by appending retrieved documents to the prompt [1]. This *static retrieval* is fundamentally different from SAMR. RAG systems index a static knowledge base and pull in chunks as needed [1], but they do not manage or evolve memory themselves. In RAG, the LLM never *decides* what to remember; all persistence lives in an external database. In contrast, our approach internalizes memory control: the agent *actively filters* what to store via SAMR, and retrieval is context-driven (e.g. SAMR can pull from its memory store as needed). Recent "agentic RAG" works do allow dynamic query planning [13], but even these keep a fixed knowledge repository. Our system's memory *grows and restructures* over time under the agent's own guidance, a deeper form of agency.

- **MemGPT and OS-inspired memory:** Packer et al. propose MemGPT, which treats the LLM like an operating system managing memory tiers [3]. MemGPT maintains caches and uses interrupts to swap memory in and out of the context [3]. This is very close in spirit. Our SAMR could serve as a "kernel" strategy: like MemGPT, we hierarchically move data between working memory (context) and slower memory store, but in our case the LLM itself drives those moves with SAMR rules. MemGPT emphasizes system calls and control flow; we emphasize continuous decision-making about relevance. Both aim for extended context, but SAMR is more fine-grained (token-level management) and streaming-oriented.

- **Long-Context Transformers:** Some work simply extends context windows (e.g. LLaMA variants to 32K/128K tokens) or uses techniques like "attention sinks" [5]. StreamingLLM shows that with a fixed

window, one can slide over very long text by keeping only recent tokens and a few key ones [5]. These methods do improve single-session length, but do not create persistent memory beyond what fits. Our framework can incorporate larger windows if available, but its novelty is in active memory decisions, not window size. In fact, SAMR could even leverage future long-window models to allocate which parts of the large window to free.

- **Continual Learning and Lifelong LLMs:** Some works consider fine-tuning models over time or using RL to update knowledge [7]. For example, Memory-R1 uses RL to teach an LLM *how* to manage its memory bank [8]. These focus on altering weights or off-line memory structures. Our SAMR, by contrast, acts *at inference time* with minimal or no fine-tuning. (In principle SAMR rules could be learned with RL, akin to Memory-R1, but that is an optional layer.) We emphasize *run-time memory control* rather than offline weight changes, making the agent adaptive without retraining.

- **LLM Agents and Tools:** Current LLM agent frameworks (AutoGPT, BabyAGI, etc.) employ retrieval tools, planners, and static memory logs [14] [15]. Our system can be viewed as a superset: the "tools" here are SAMR and streaming. Unlike typical prompt-based agents, which rely on fixed prompt engineering or pre-saved memory entries, our agent continuously introspects and updates itself. In other words, we do not rely on a rigid prompt for behavior; the system's behavior emerges from the feedback between SAMR, memory, and input. This contrasts with static prompt–response models, which lack any notion of evolving state.

Prototype Architecture

A conceptual architecture for our system is shown below. Input sensors (microphone, camera, text interface) continuously feed data into a **Preprocessing** stage, which tokenizes or encodes raw signals (e.g. speech-to-text, object detection to tokens). These tokens flow into the **LLM Core**'s context window. Surrounding the core is a **SAMR controller** that monitors context usage. A **Memory Manager** maintains an external episodic memory store (e.g. a vector database or key-value store). The LLM, SAMR, and memory manager form a loop:

- The LLM processes the current context and produces outputs or actions.
- The SAMR controller analyzes new context tokens: it may invoke the memory manager to *save* summaries of parts of context, or to *retrieve* relevant past memories back into context.
- The memory manager updates its entries based on SAMR's instructions (add new facts, update old ones, delete unneeded data).

This loop repeats continuously. In each iteration, the model's response is a function of *all* tokens in its context (recent input + any recalled memory). Because the agent is live, the "Output/Actions" may also influence future input (e.g. speaking yields new audio to hear). Over time, the memory store grows a graph of important concepts and experiences, while the context window slides forward.

(Figure 1: Conceptual architecture of the SAMR + streaming AGI system, showing sensor input feeding into the LLM, with the SAMR loop managing a persistent memory store and updating the LLM's context.)

Theoretical Implications and Limitations

Our framework shifts several theoretical points:

- **Context Window as Lifespan:** The agent's immediate memory span is finite. If it truly had infinite "life," its context window would need to grow without bound. Instead, we accept that *after N tokens*, the agent must compress or forget earlier content. Thus, the *effective lifespan* of any given memory in the agent's working memory is the window size. This is an inherent limitation: crucial details may be irrevocably lost if SAMR fails to capture them.

- **Tradeoff of Model vs Memory:** The agent can either put knowledge into weights (via training or fine-tuning) or into its contextual/memory system. Our proposal largely avoids weight changes, trading off *online adaptability* for *static knowledge*. In other words, we gain flexibility and continuity at the cost of not continuously learning new tasks at the parameter level. This is acceptable for many applications (an assistant that improves over one session) but not a replacement for lifelong learning research [16].

- **Scalability:** The memory manager may become large. While an LLM's context is fixed, the external memory is not. Mechanisms are needed to prune or compress that memory (for instance, regularly summarizing old memory entries). Without such measures, storage grows without bound. Our SAMR implicitly provides some pruning by deciding what to delete or merge. Future systems could incorporate memory distillation: periodically, the agent could retrain itself on summarized experiences, distilling memory into its own parameters and thus freeing memory.

- **Emergent Behavior:** Scientifically, we do not claim that SAMR + streaming yields true human-level AGI. Rather, it endows LLMs with structured persistence. It allows identity and knowledge to evolve, which are necessary but not sufficient for AGI. The true intelligence will still be limited by the model's reasoning capacity and training. However, by bridging the gap between episodic LLMs and continuous agents, we provide a platform to study multi-turn cognition.

Future Work

This framework opens many directions. One is **memory distillation**: periodically leveraging the LLM to review and compress its memory store into high-level schemas or even to fine-tune itself, akin to a student studying notes. Another is **lifelong learning**: enabling not only memory updates but weight updates based on memory (e.g. using RLOptimizer or evolutionary methods). We also foresee **modular updates**: instead of one monolithic model, different sub-networks could specialize (language, vision, motor), and SAMR-like controllers could route information between them. Finally, advancing the streaming front-end (better audio/video tokenizers) will make the agent more robust in real environments.

Conclusion

We have outlined an AGI framework defined by *temporal continuity, context-bounded memory, and dynamic recall*. By redefining intelligence as a process with a moving window of awareness, we motivate the SAMR mechanism and live-stream data integration. SAMR lets the LLM decide what to remember, forget, and compress, addressing the finite context window issue [4] [7]. Streamed tokens ensure the agent

continuously ingests new information, as shown feasible in multi-modal research [9] [10]. Together, these yield an agent that exhibits proto-AGI traits (consistent identity, resilience, task-flexibility) beyond any static prompt–response model.

Compared to prior work – RAG's passive retrieval [1] [2], MemGPT's OS-like memory tiers [3], or window-scaling transformers [5] – our system is distinguished by its live, agentic memory loop. We have proposed a conceptual prototype architecture and identified the tradeoffs (notably, the context window still bounds memory lifespan). With citations to current literature for each component, this framework aims to inspire AI researchers and engineers to explore continuous, memory-augmented LLM agents as stepping stones to AGI.

Sources: The concepts and comparisons above are supported by recent LLM-memory literature: for example, LLMs' statelessness [4], RL-managed memory operations [7] [8], RAG methods [1] [2], MemGPT's OS analogies [3], and streaming multimodal LLMs [9] [10]. Each source is cited where relevant to ensure scientific grounding.

---

[1] [11] [12] [13] A-Mem: Agentic Memory for LLM Agents
https://arxiv.org/html/2502.12110v11

[2] [2005.11401] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks
https://arxiv.org/abs/2005.11401

[3] [2310.08560] MemGPT: Towards LLMs as Operating Systems
https://arxiv.org/abs/2310.08560

[4] [7] [8] Memory-R1: Enhancing Large Language Model Agents to Manage and Utilize Memories via Reinforcement Learning
https://arxiv.org/html/2508.19828v4

[5] [6] GitHub - mit-han-lab/streaming-llm: [ICLR 2024] Efficient Streaming Language Models with Attention Sinks
https://github.com/mit-han-lab/streaming-llm

[9] Continuous Speech Tokens Makes LLMs Robust Multi-Modality Learners
https://arxiv.org/html/2412.04917v1

[10] [2405.16009] Streaming Long Video Understanding with Large Language Models
https://arxiv.org/abs/2405.16009

[14] Introduction to LLM Agents | NVIDIA Technical Blog
https://developer.nvidia.com/blog/introduction-to-llm-agents/

[15] Effective context engineering for AI agents \ Anthropic
https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents

[16] Understanding the Limits of Lifelong Knowledge Editing in LLMs
https://arxiv.org/html/2503.05683v1