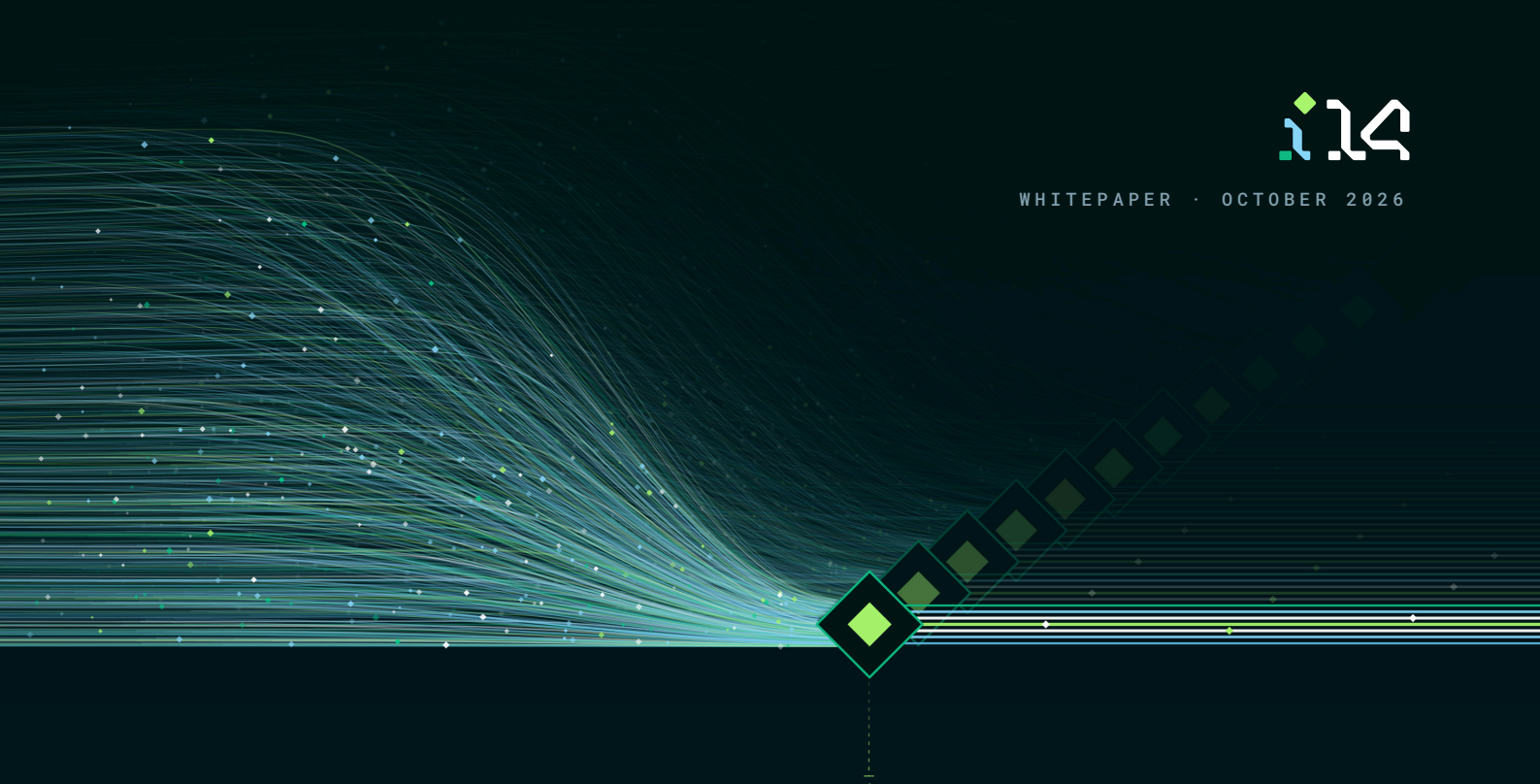
WHITEPAPER · OCTOBER 2026



# Sugar Glider

Constant-memory language models that read a million tokens, with a 32× lighter memory that can be resized to fit the device

Rivaan Patil  Simon Dennis  Kevin Shabahang  Hao Guo  i14.ai

◆ ABSTRACT

We introduce Sugar Glider, a family of language models whose memory and per-token compute are constant, $\mathcal{O}(1)$ in context length. Built on NVIDIA Nemotron 3 Nano 4B and Nemotron 3 Super 120B-A12B, Sugar Glider matches its parents on knowledge, code, reasoning, and mathematics benchmarks. At one million tokens, Sugar Glider-120B needs 1/32 of its parent's memory, serving 32–36× more concurrent requests: 453 versus 14 on four H200s. Because the memory budget is set at inference time, one set of weights serves small and large deployments alike, and even at four times its default budget Sugar Glider-120B fits 10× more million-token requests than its parent. On the full 13-task RULER at one million tokens, Sugar Glider-120B scores 73.3, the strongest result for any $\mathcal{O}(1)$ model, and Sugar Glider-4B achieves perfect single-needle retrieval from 32k to 1M tokens.

**32×**
as many million-token requests per server as Nemotron 3 Super

**$\mathcal{O}(1)$**
memory per sequence, at any context length

**any size**
memory sized at run time, one set of weights

◆ The latest version of Sugar Glider is built on Cerbearus, our proprietary underlying architecture.

01 /

# Introduction

On one server of four H200 GPUs, NVIDIA Nemotron 3 Super 120B-A12B [3] fits 14 million-token requests at once. Sugar Glider-120B, built on it, fits 453 on the same server, 32 times as many, and maintains parity with it on knowledge, code, reasoning and mathematics.

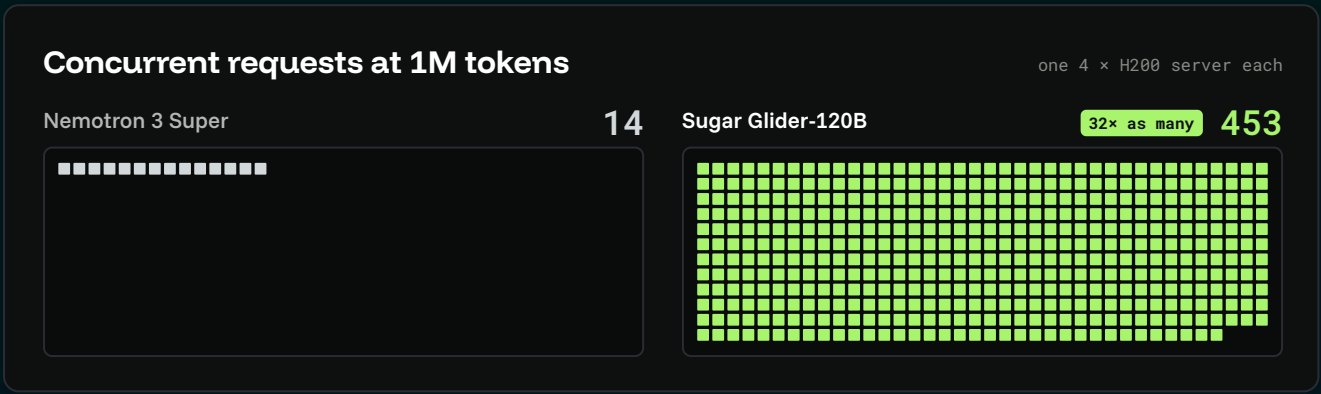



FIGURE 1 **One server, four H200 GPUs.** Each square is a million-token request held in memory at once.

The difference is memory. Attention gives a transformer an exact memory of everything it has read, at a price that grows with every token: every attention layer keeps the keys and values of every earlier token in a cache that grows by one entry per token [4]. Reading a million tokens means carrying a million tokens' worth of keys and values in every attention layer, so how far a model can read, and how many readers a server can hold at once, are set as much by the device as by the model.

Sugar Glider removes that dependence. Its memory has a fixed size: the memory it carries from one token to the next, and the work it does per token, stay constant however long the context grows, so the model is $\mathcal{O}(1)$ in the length of its context. It reads a context as a stream, and at a million tokens it carries the same memory as at the first, so what a deployment must provision no longer depends on how long its inputs are. The memory's size is chosen when the model runs: one set of weights runs with a compact memory where memory is scarce and a large one where it is plentiful.

Our contributions are:

- **32 times the million-token requests.** On four H200s, Sugar Glider-120B fits 453 million-token requests to Nemotron 3 Super's 14; on four H100s, 73 to 2, 36 times as many (Section 2).
- **Parity with the parent.** Built on NVIDIA Nemotron 3 Nano 4B and Super 120B-A12B [2, 3],[1] both models match their parents on knowledge, code, reasoning and mathematics (Section 3).
- **The strongest $\mathcal{O}(1)$ RULER score at a million tokens.** Sugar Glider-120B scores 73.3 on the full 13-task RULER at 1M, the strongest result for any $\mathcal{O}(1)$ model; Sugar Glider-4B answers every single-needle question from 32k to 1M (Section 3).
- **One set of weights, any memory.** The memory's size is chosen at run time; at a million tokens, eight times Sugar Glider-4B's default memory adds only 0.6 points (Section 4).

[1] nvidia/NVIDIA-Nemotron-3-Nano-4B-BF16 and nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-BF16.

02 /

# Thirty-two times the million-token requests

**Memory sets how many requests a server can hold.** A server runs many requests at once, and every one of them holds its memory until it finishes. A conventional model's requests grow with their context, so the longer they are, the fewer fit: on one server of four H200 GPUs, Nemotron 3 Super fits 109 requests of 128k tokens, 56 of 256k and only 14 of a million. Sugar Glider-120B's requests do not grow, so the same server fits 453 of them at every one of those lengths (Figures 1 and 2).




FIGURE 2

**32 times as many million-token requests on the same server.** On four H200 GPUs, Nemotron 3 Super fits 14 million-token requests and Sugar Glider-120B fits 453; on four H100 GPUs, 2 and 73. Requests that fit in the server's memory at once at 128k, 256k and 1M tokens, for Nemotron 3 Super, whose cache grows with every token, and Sugar Glider-120B, whose memory does not, with its memory set four times larger and at its default. Above the bars, how many times its parent's count.

03 /

# Reading to a million tokens

**Parity with the parent.** Both Sugar Glider models match their Nemotron 3 parents on knowledge, code, reasoning and mathematics benchmarks: Sugar Glider-120B fits 32 times as many million-token requests on one server as Nemotron 3 Super, and matches it in all four.

**Measured to a million tokens.** We measure long-context ability with RULER [1]: thirteen synthetic tasks that hide needles among distractors, trace variable chains, find the most common or frequent words of long lists and answer questions over long documents, at lengths from 4k to a million tokens, a context of a few thousand pages. Figure 3 shows both models.

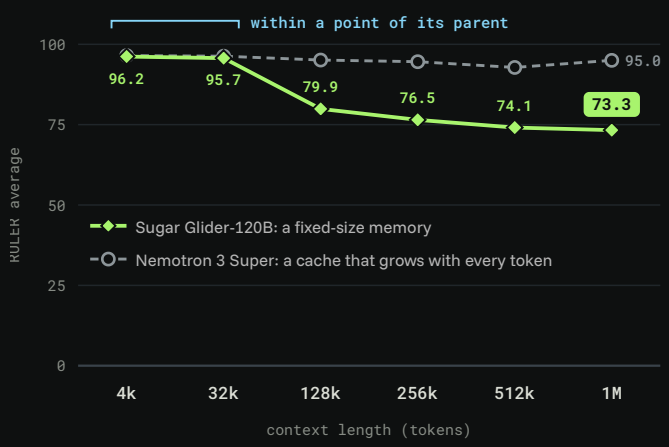
At a million tokens, Sugar Glider-120B scores 73.3 on the full 13-task RULER, the strongest result for any $\mathcal{O}(1)$ model. It is within a point of its parent at every length up to 32k tokens (95.7 against 96.3 at 32k), and it reads on to a million tokens with a memory of

one fixed size, while its parent's cache grows with every token: 79.9 at 128k, 76.5 at 256k, 74.1 at 512k and 73.3 at a million.

Sugar Glider-4B answers every single-needle question at every length from 32k to a million tokens, and finds every needle hidden among distractor needles (multi-key 1) at each of them. At a million tokens, with a memory incredibly tiny next to the cache its parent, Nemotron 3 Nano 4B, would carry, it scores 88 on variable tracking. At four times its default memory it averages 94.4 at 32k and 85.8 at 128k (Figure 3).

RULER, 4K TO 1M TOKENS

**Sugar Glider-120B and its parent**




**Scores by context length**

| | 4k | 32k | 128k | 256k | 512k | 1M |
|---|---|---|---|---|---|---|
| **Sugar Glider-120B** | **96.2** | **95.7** | **79.9** | **76.5** | **74.1** | **73.3** |
| **Sugar Glider-4B** | | | | | | |
| single-needle questions | – | **100** | **100** | **100** | **100** | **100** |
| needle among distractors | – | **100** | **100** | **100** | **100** | **100** |
| variable tracking | – | **100** | **100** | 98 | 98 | 88 |
| 13-task average | – | 94.4 | 85.8 | 74.7 | 66.6 | 61.0 |

FIGURE 3

**73.3 at a million tokens, the strongest RULER score for any $\mathcal{O}(1)$ model.** Left: the RULER average from 4k to 1M tokens for Sugar Glider-120B and its parent, Nemotron 3 Super 120B-A12B, whose attention cache grows with every token. Right: the same scores, and Sugar Glider-4B's from 32k to 1M tokens: single-needle questions, a needle among distractor needles (multi-key 1), variable tracking and the average of all 13 tasks.

04 /

# One set of weights, any memory

**A memory that can be resized.** The memory's capacity is a setting of the run, chosen to fit the device, and nothing is retuned between sizes: one set of weights serves a compact memory on a small device and a large one in a data center. The setting is also a dial on how many requests a server holds (Figure 2).

**A small memory goes a long way.** Figure 4 runs Sugar Glider-4B over 128k tokens of context with memory settings from half its default to four times it. Even with half its default memory, the model answers every single-needle question and averages 71.3; its default memory gives 73.4, and four times the default 85.8. At 128k tokens, a larger memory recalls more, and the choice is made at run time.

**The default memory reads a million tokens.** At a million tokens the memory's size barely matters (Figure 4, right). Sugar Glider-4B scores 60.4 with its default memory, 61.0 with four times it and 61.0 with eight times: an eightfold larger memory adds just 0.6 points. Sugar Glider does not need a bigger memory to read further: its default memory, incredibly tiny next to its parent's cache at a million tokens, is already enough.

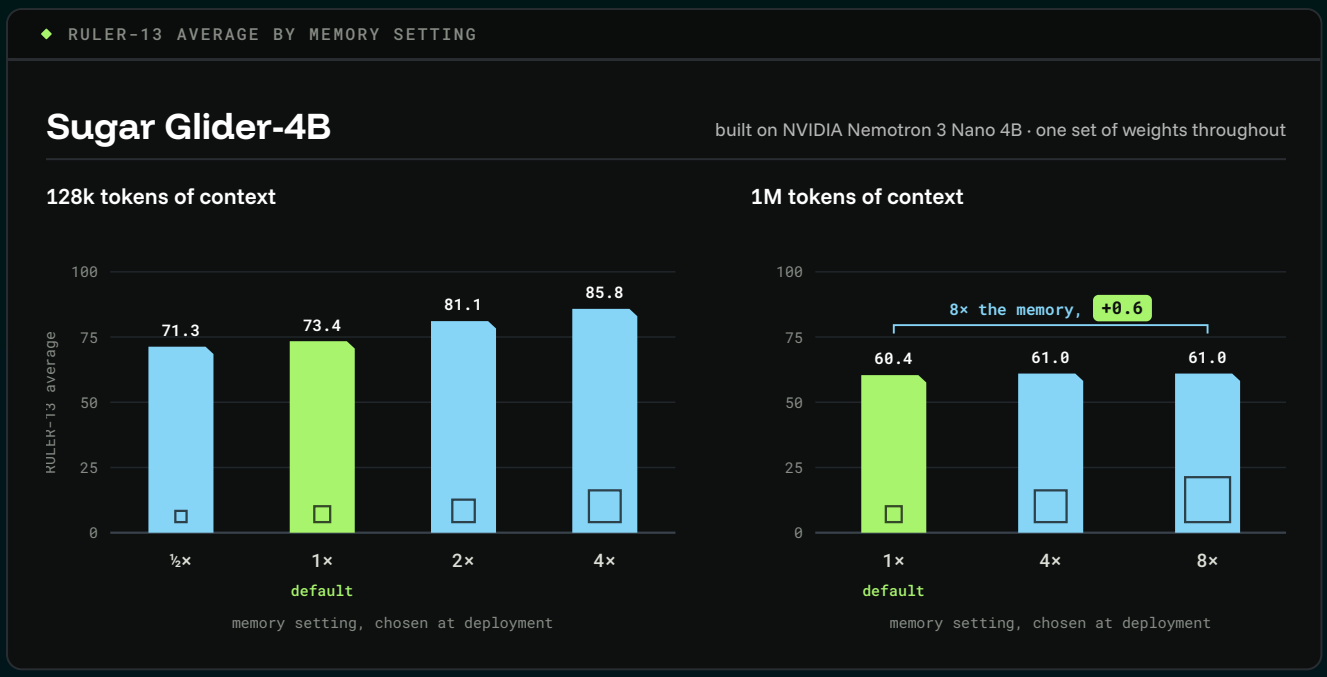



FIGURE 4

**A small memory goes a long way.** Sugar Glider-4B at several memory settings, one set of weights throughout. Left: at 128k tokens, even half the default memory answers every single-needle question, and four times it, the setting of Figure 3, reaches 85.8. Right: at a million tokens, an eightfold larger memory adds only 0.6 points to the default (RULER-13 average).

**Choosing a memory.** The default fits the most requests, and at a million tokens an eightfold larger memory adds only 0.6 points to it. Where recall at 128k tokens matters most, four times the default raises Sugar Glider-4B's score there from 73.4 to 85.8, and at that setting Sugar Glider-120B fits ten times its parent's million-token requests; where memory is scarce, even half the default answers every single-needle question at 128k tokens.

05 /

# What a memory that never grows makes possible

**Whole documents, whole codebases.** A million tokens is a few thousand pages: a contract with every amendment and exhibit, a year of a team's messages, a large codebase with its history. Sugar Glider reads all of it with the same fixed memory, so the length of what it is asked to read is no longer a question of what the device can hold.

**Agents that keep working.** An agent that plans, calls tools and reads their results gathers context with every step. With a conventional transformer its memory grows until the session must be cut short or its history condensed to make room. Sugar Glider's memory stays the same size while the agent works: with a million tokens of history it carries the same memory as at the first.

**Long context on small devices.** Because the memory is small and fixed, and its size is a setting of the deployment, reading a long context is no longer reserved for the largest accelerators. On four H100 GPUs, where Nemotron 3 Super fits two million-token requests, Sugar Glider-120B fits 73.

**Predictable serving.** With a fixed memory per sequence, the memory each request needs is known before its first token arrives, whether the request is a paragraph or a book. Long requests no longer claim memory in proportion to their length: four H200 GPUs hold 453 Sugar Glider-120B requests at 128k tokens and 453 at a million, so capacity can be planned per request rather than per token.

**One model for every device.** A single set of weights spans small and large deployments. Model builders evaluate and ship one checkpoint, and each deployment chooses its memory, instead of maintaining a separate model for each memory budget.

06 /

# Conclusion

Sugar Glider-120B fits 32 times as many million-token requests as its parent on the same server: on four H200 GPUs, 453 where Nemotron 3 Super fits 14, and on four H100 GPUs, 73 where it fits 2. Built on NVIDIA Nemotron 3, both Sugar Glider models match their parents on knowledge, code, reasoning and mathematics while carrying a memory that never grows. Sugar Glider is $\mathcal{O}(1)$ in the length of what it reads, and on the full RULER at a million tokens Sugar Glider-120B scores 73.3, the strongest result for any $\mathcal{O}(1)$ model. The memory's size is chosen by whoever runs it, so one set of weights fits small and large devices, and at a million tokens an eightfold larger memory adds only 0.6 points to Sugar Glider-4B's default. With a memory that never grows, the length a model can read is no longer set by its hardware, and a server holds as many million-token requests as it does short ones.

## References


[1] Cheng-Ping Hsieh et al. **RULER: What's the real context size of your long-context language models?** In COLM, 2024.

[2] NVIDIA. **NVIDIA Nemotron 3: Efficient and open intelligence.** arXiv preprint arXiv:2512.20856, 2025.

[3] NVIDIA. **Nemotron 3 Super: Open, efficient Mixture-of-Experts hybrid Mamba-Transformer model for agentic reasoning.** Technical report, NVIDIA, 2026.

[4] Ashish Vaswani et al. **Attention is all you need.** In NeurIPS, 2017.