

EdgeRock: 长程推理，恒定内存

Huishan Ji
jihs@stonehill-tech.com

Weiming Zhang
zhangwm@stonehill-tech.com

Qirui Ye
qiruiy@andrew.cmu.edu

Jinchao Xu
jinchao.xu@kaust.edu.sa

Yike Guo
yikeguo@ust.hk

Hua Xu^{*}
huaxu@ust.hk

 github.com/rockmatrix-ai/EdgeRock

 huggingface.co/rockmatrix-ai/EdgeRock

摘要

我们推出 EdgeRock 系列多模态模型，在恒定内存下实现强大的推理能力。如今，大多数模型把推理历史存在键值 (KV) 缓存中，每生成一个 token 就多存一些。于是推理越长，占用的加速器内存越多，能同时服务的请求越少；到了端侧，还要与感知、控制争抢有限的内存。反之，若历史的容量固定，内存就能一次分配到位，推理再长也不会耗尽。不过，恒定内存本身并不新鲜：循环模型、状态空间模型和滑动窗口模型等有界状态模型早已做到这一点，却往往牺牲了推理能力，因为它们留不住后续推理要用的远处细节。为此，我们引入记忆岩锚 (Rock Memory Anchors, RMA)：模型学会挑出值得记住的内容，写成 KV 记录存入固定容量的锚点库，即使原文相隔很远，仍可直接读取。首个开源模型 EdgeRock-4B 支持文本与图像推理，AIME 2024–2026 平均一次通过率 (pass@1) 达 82.9%，而我们评测的有界状态模型即便达到 14B 规模，最好也只有 18.8%。这一成绩已与内存随推理增长的领先小尺寸模型处于同一水平。同时，与转换前的源模型相比，EdgeRock-4B 的 KV 缓存最多减少 99.1%，端到端批量吞吐量最高提升至 6.42 倍。

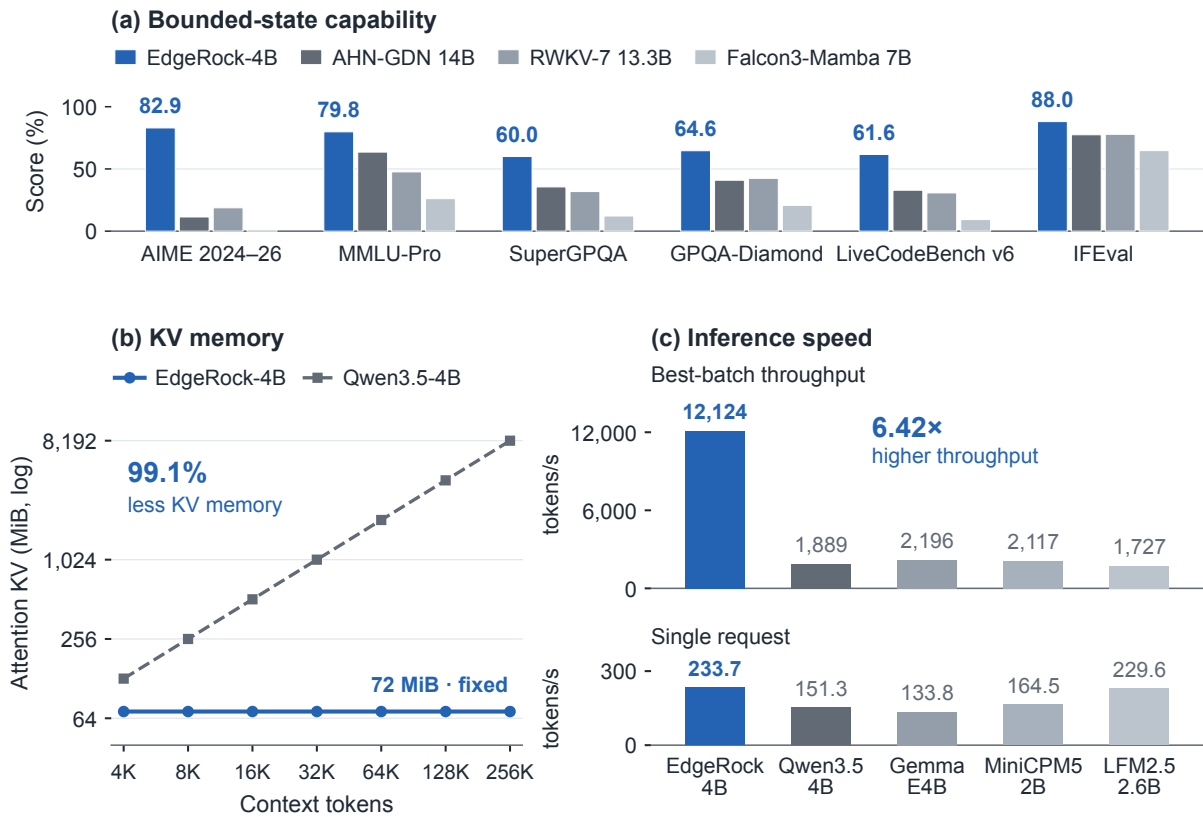


图1 恒定内存下的推理能力。 (a) 在图示六项文本指标上，EdgeRock-4B 均领先于参与比较的 7B–14B 有界状态模型；AIME 对 2024–2026 三届成绩等权平均。 (b) 按配置计算的单序列 BF16 注意力 KV 容量，坐标采用对数刻度。EdgeRock 固定使用 72 MiB，在 256K 长度下比 Qwen3.5-4B 少 99.1%。 (c) 上图：重复提示词、32K 输入与 32K 输出时，各模型在最佳实测批大小下的输出吞吐量（附录表 19）。仅统计输出 token，耗时包含预填充；EdgeRock 达到 Qwen3.5-4B 的 6.42 倍。下图：128K 输入、128K 输出、批大小为 1 时的累计解码速率。测试均在单张 Hopper 80GB GPU 上使用 BF16 权重与 KV。

^{*}通讯作者。

1 引言

长推理让内存成为模型部署的瓶颈。采用全注意力时 [1]，模型会把每个 token 的键和值都存进键值 (KV) 缓存，因此推理 100K 个 token，就要存下 100K 个 token 的历史。本文将这类内存随推理增长的模型称为历史增长型模型。在数据中心，这些缓存挤占有限的高带宽内存 (HBM)，单个加速器能同时服务的请求随之减少，服务成本则水涨船高 [2]。

更关键的是，许多部署需要的不是更小的缓存，而是固定的缓存。在端侧，机器人和车辆要在有限内存中为感知、规划与控制预留资源 [3]；专用芯片的片上容量更是在设计时就已确定。缓存一旦持续增长，系统就只能二选一：要么按最长轨迹预留空间，要么在内存耗尽时提前结束推理。固定历史容量让资源分配有了确定的边界：同一序列处理 4K 或 256K 个 token，历史状态使用的内存相同。这些场景需要效率，更需要确定性。

近年的架构已大幅压缩 KV 缓存：多头潜在注意力 (MLA) 压缩每个 token 的缓存表示 [4]；Qwen3.5 与 Kimi Linear 分别以 Gated DeltaNet (GDN) 和 Kimi Delta Attention (KDA) 为主，每四层只保留一层全注意力 [5, 6]；DeepSeek 稀疏注意力 (DSA) 只读取部分条目 [7]，DeepSeek-V4 的压缩稀疏注意力 (CSA) 等机制进一步减少需要存储的条目 [8]；KIVI 等 KV 量化方法则降低每个条目的位数 [9]。不过，这些方法只降低了单个 token 的缓存成本，模型每读入或生成一个 token，缓存仍会增长。减缓增长只能推迟触及上限，却无法消除这个上限。

另一条路线是有界状态模型，它们的历史存储本来就是固定的：循环模型和状态空间模型将历史压缩进定长状态 [10, 11, 12]，滑动窗口注意力及其与循环层的混合架构则只保留近期 token [13, 14, 15]。问题在于，我们评测的 16 个有界状态模型都没有达到实用的推理水平：即使参数规模达到 14B，AIME 2024–2026 成绩也没有超过 18.8% (第 6 节)。推理时淘汰 KV 也能给历史增长型模型的缓存设上限，例如只保留注意力汇聚位置、近期 token 或得到过较高注意力的 token [16, 17, 18]；但这些模型训练时从未学过只凭剩下的内容推理。

症结在于，长推理需要反复回到数千个 token 之前的细节：一个定义、题目中的一项约束，或某一步的中间结果。这些信息一旦移出滑动窗口，便再也读不到；循环状态则只留下压缩摘要，不同细节容易混在一起 [19]。问题由此变成：能否在固定内存中留住后续推理要用的远处细节，让恒定内存不再以牺牲推理为代价？

EdgeRock 用记忆岩锚 (Rock Memory Anchors, RMA) 解决这一问题。它的设计取意于岩锚：将推理所需的关键信息固定下来。每个记忆岩锚 (下文简称锚点) 是一条学习得到的键值记录，保存在固定容量的锚点库中。三部分各司其职：循环状态概括不断演变的上下文，局部注意力覆盖最近 1,024 个 token，RMA 则从更早的历史中留住最有价值的细节。取舍过程如下：写入器为每个 token 生成一条候选记录，评分网络给它打分；每累积 256 条候选，就与锚点库中已有的记录一起比较，只留下得分最高的 1,024 条 (各 KV 组分别进行)，其余淘汰。留下的记录彼此独立、内容不变，因此即使来源早已移出窗口，后续查询仍可直接读取。局部注意力与 RMA 把每条序列的注意力 KV 固定在 72 MiB；加上循环状态，一条序列需保存的全部历史，即持久历史状态，在批量运行时固定为 97.76 MiB。

首个开源模型 EdgeRock-4B 表明，恒定内存与强大推理可以兼得 (表 1)。

表 1 EdgeRock-4B 的核心结果。

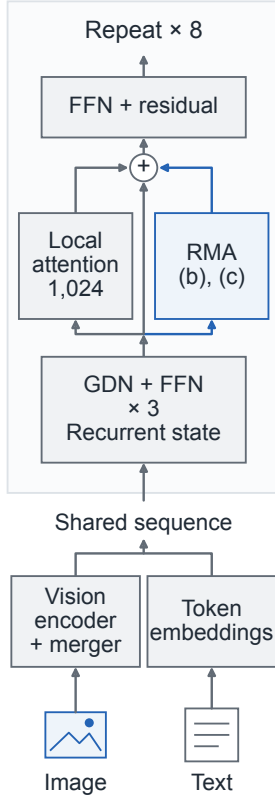
核心主张	实现方式	实验证据
恒定内存	循环状态、1,024 个 token 的局部窗口和固定容量的 RMA 锚点库。	每条序列的注意力 KV 始终为 72 MiB；在 256K 上下文下比 Qwen3.5-4B 少 99.1% (第 4 节)。
实用推理能力	写入、准入与检索经分阶段训练习得。	AIME 2024–2026 达到 82.9%，16 个有界状态对照模型最高为 18.8%；进入领先小尺寸模型的 82.2–87.8% 区间 (第 6 节)。
RMA 的关键作用	来源移出窗口后，锚点仍可被单独检索。	推理时关闭 RMA，长上下文查询准确率从 94.1% 降至 0.4% (第 7 节)。
更高服务效率	每条序列的历史固定：每个 token 的计算量不随上下文增长，单卡可同时运行更多序列。	单张 Hopper 80GB GPU 并发运行 512 条长生成序列；吞吐量平均达到 Qwen3.5-4B 的 4.94 倍，最高 6.42 倍 (第 8.1 节)。

2 如何构建 EdgeRock ?

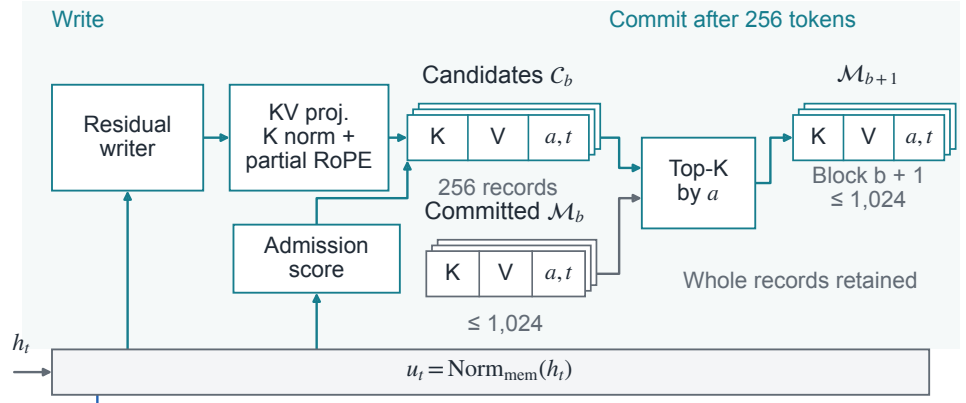
2.1 解码器结构

EdgeRock-4B 基于 Qwen3.5-4B 语言骨干进行改造 [5]。模型保留 24 个 Gated DeltaNet (GDN) 层，将八个全注意力层替换为覆盖 1,024 个位置的滑动窗口注意力 [13] 与负责保留远距离信息的独立 RMA 模块。解码器

(a) Hybrid decoder



(b) Write and select



(c) Retrieve and integrate

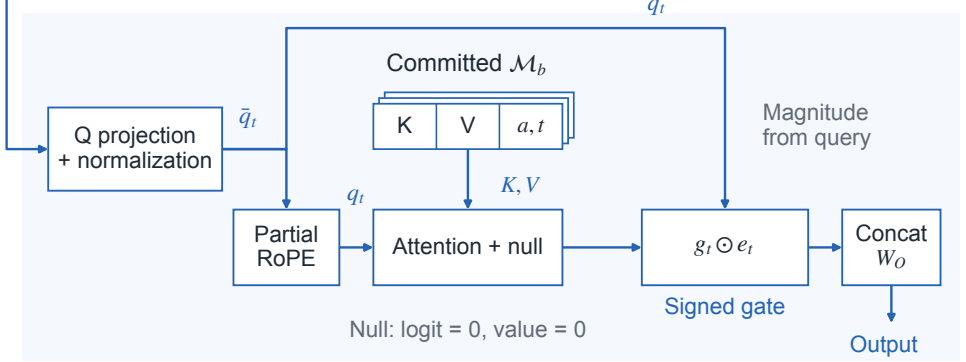


图2 EdgeRock 架构。(a) 文本与图像特征进入同一个解码器，由循环状态、局部注意力和 RMA 共同处理。(b) 每个 KV 组从已提交锚点库与写入器新生成的 256 条候选记录中，选择最多 1,024 条记录；更新后的记录在下一块生效。(c) 当前查询在已提交锚点与空注意力项之间分配权重，再由查询相关的有符号门控调节记忆输出。记忆输出与局部注意力输出共同进入残差流。

的层排列为

$$[\text{GDN}, \text{GDN}, \text{GDN}, \text{SWA}_{1024} \parallel \text{RMA}] \times 8. \quad (1)$$

每个序列混合模块后接一个前馈子层；八个 RMA 模块各自拥有独立的参数与状态。图像经视觉编码器和合并模块处理后，与文本嵌入组成统一的解码器输入序列（图 2a）。

GDN 使用固定维度的循环矩阵 [20]。局部窗口与锚点库分别存储 KV 记录；同一个来源位置可以同时出现在两者之中。

2.2 容量固定，内容自适应

EdgeRock 始终保持持久历史状态的容量不变，只更新其中的内容。用 $t = 0, 1, \dots$ 表示因果序列位置，用 $b(t) = \lfloor t/C \rfloor$ 表示当前位置所属的候选块，每块包含 $C = 256$ 个位置。在处理位置 t 之前，状态与转移可写为

$$\mathcal{H}_t = (\mathcal{R}_t, \mathcal{L}_t, \mathcal{M}_{b(t)}, \mathcal{C}_t, \pi_t), \quad (2)$$

$$(y_t, \mathcal{H}_{t+1}) = F_\theta(x_t, \mathbf{p}_t, \mathcal{H}_t). \quad (3)$$

其中， x_t 为输入嵌入， y_t 为解码器输出， $\mathbf{p}_t$ 为旋转位置坐标。 $\mathcal{R}_t$ 汇集解码器中的循环状态， $\mathcal{L}_t$ 表示局部 KV， $\mathcal{M}_{b(t)}$ 表示已提交的锚点库， $\mathcal{C}_t$ 表示待提交候选记录， π_t 表示进度元数据。

2.3 检索结果融入推理

省略层索引后，每个局部注意力与 RMA 层将两路信息共同写入残差流：

$$z_t = h_t + \text{SWA}(\text{Norm}_{\text{local}}(h_t)) + \text{RMA}(h_t, \mathcal{M}_{b(t)}), \quad (4)$$

$$h_t^{\text{out}} = z_t + \text{MLP}(\text{Norm}_{\text{ffn}}(z_t)). \quad (5)$$

RMA 的归一化、写入、准入评分，以及查询、键、值和输出投影都使用独立参数。残差 z_t 把局部信息与检索结果一起传给后续层。

表 2 EdgeRock-4B 配置。

配置项	数值	配置项	数值
模型规模	4B	RMA 参数量	372M
循环层 / 局部注意力与 RMA 层	24 / 8	残差维度	2,560
注意力查询头 / KV 组数	16 / 4	注意力头维度	256
局部窗口 W	1,024	已提交记录容量 K	1,024
候选块大小 C	256	RMA KV 精度	BF16

2.4 分阶段转换

EdgeRock-4B 由 Qwen3.5-4B 经四个阶段转换而来。RMA 适配 (M0) 让新增的 RMA 模块学会保留和检索后续预测所需的信息。联合适配 (R) 将 RMA 与循环、局部注意力骨干共同训练，让后续层学会使用检索结果。监督后训练 (S) 基于适配后的历史机制训练通用回答能力。随后，可验证奖励强化学习 (RLVR) [21, 22] 再根据答案正确性与任务完成情况改进输出。第 7 节展示各阶段的记忆准确率 (图 5a)。

3 RMA 如何保留远距离信息？

处理每个 token 时，RMA 读取已提交锚点库，并用同一输入生成候选记录。到达块边界时，准入机制从旧锚点与新候选中选择记录，更新后的锚点库从下一块起可读。候选累积期间，近期 token 之间的交互由局部注意力负责。完整的逐 token 流程见附录 E 中的算法 1。

3.1 独立的记忆写入器

记 $u_t = \text{Norm}_{\text{mem}}(h_t) \in \mathbb{R}^D$ ，其中 $D = 2,560$ 。独立的残差写入器采用 SwiGLU 式门控投影 [23]，生成适合存储的表示：

$$w_t = u_t + W_{\text{down}}[\text{SiLU}(W_{\text{gate}}u_t) \odot W_{\text{up}}u_t]. \quad (6)$$

RMA 的 KV 投影以写入器输出 w_t 为输入，查询投影以 u_t 为输入。写入器的中间维度为 2,560 [24, 25]。

3.2 四个 KV 组分别因果准入

一个小型网络为每个 KV 组计算保留优先级：

$$a_t = W_{a,2} \text{SiLU}(W_{a,1}u_t), \quad a_t \in \mathbb{R}^4. \quad (7)$$

评分网络由无偏置的 $2560 \rightarrow 256 \rightarrow 4$ 投影组成，根据当前因果上下文产生 FP32 分数 [24, 25]。各组独立保留得分最高的记录，不同组可以选中相同的来源位置。

3.3 分块提交与因果可见性

每经过 256 个 token，已提交锚点与新候选记录共同竞争每个 KV 组的 1,024 个槽位。

对于候选块 $\mathcal{B}_b = \{bC, \dots, (b+1)C - 1\}$ 、KV 组 g ，取容量 $K = 1024$ 、块大小 $C = 256$ ，有

$$C_{b,g} = \{(k_{t,g}, v_{t,g}, a_{t,g}, t) : t \in \mathcal{B}_b\}, \quad (8)$$

$$\mathcal{M}_{b+1,g} = \text{TopK}_{a_g}(\mathcal{M}_{b,g} \cup C_{b,g}; K). \quad (9)$$

在块 b 内，查询只读取 $\mathcal{M}_{b,g}$ ；本块选中的候选记录到块 $b+1$ 才可读取。候选缓冲区累积 256 个 token 的同时，覆盖 1,024 个位置的局部窗口处理块内交互。每次更新在每组内最多检查 $K + C = 1280$ 条记录，随后复用候选缓冲区。

算子 $\text{TopK}(A; K)$ 最多保留 K 条记录，先按分数从高到低排序，分数相同则优先保留因果位置更早的记录。由于分数在写入时即已确定，已提交锚点库满足以下精确不变量：

$$\mathcal{M}_{b,g} = \text{TopK}_{a_g}\left(\bigcup_{r=0}^{b-1} C_{r,g}; K\right), \quad \mathcal{M}_{0,g} = \emptyset. \quad (10)$$

每条被淘汰的记录都排在至少 K 条更高优先级记录之后，新增候选不会使它重新进入全局前 K 。因此，容量有限的增量更新始终保留了所有已完成块中优先级最高的记录。容量限制的是记录条数，而不是保留时长：只要未被新记录替换，原记录就始终可直接访问。

3.4 按查询检索锚点库

后续查询可以访问任意仍被保留的记录，即使其来源早已离开局部窗口。每个模块使用 16 个查询头、四个 KV 组 [26]，头维度 $d = 256$ 。查询与键采用带可学习增益的 RMS 归一化 [27, 24, 25]：

$$\text{RMS}_\gamma(x) = \frac{\gamma \odot x}{\sqrt{d^{-1} \sum_{i=1}^d x_i^2 + 10^{-6}}}. \quad (11)$$

查询由当前位置的表示 u_t 生成，键和值则由写入器输出 w_t 投影得到：

$$\bar{q}_{t,h} = \text{RMS}_{\gamma_Q}(W_{Q,h}u_t), \quad q_{t,h} = R(\mathbf{p}_t)\bar{q}_{t,h}, \quad (12)$$

$$k_{t,g} = R(\mathbf{p}_t)\text{RMS}_{\gamma_K}(W_{K,g}w_t), \quad v_{t,g} = W_{V,g}w_t. \quad (13)$$

各投影满足 $W_{Q,h}, W_{K,g}, W_{V,g} \in \mathbb{R}^{d \times D}$ 。查询与键的增益参数相互独立，在同一模块的各头或各组之间共享，有效增益为 $\gamma = 1 + \omega$ 。部分旋转变换 [28] $R(\mathbf{p}_t)$ 仅作用于 256 个坐标中的 64 个，基数为 10^7 ；值既不进行该变换，也不采用 Q/K 归一化。纯文本输入使用 $\mathbf{p}_t = (t, t, t)$ ，多模态输入沿用骨干的时空坐标（附录 E）。

对于属于 KV 组 $g(h)$ 的查询头 h ，用 $I_{b,g}$ 索引 $\mathcal{M}_{b,g}$ 中的记录，并定义 $\ell_{tj,h} = q_{t,h}^\top k_{j,g(h)}/\sqrt{d}$ 。令 $b = b(t)$ ，从已提交锚点库读出的结果为

$$e_{t,h} = \sum_{j \in I_{b,g(h)}} \frac{\exp(\ell_{tj,h})}{1 + \sum_{i \in I_{b,g(h)}} \exp(\ell_{ti,h})} v_{j,g(h)}. \quad (14)$$

每个头最多读取 1,024 条记录。查询头共享对应组的 KV 库，但各自计算注意力权重。

检索内容与强度。 空项的 logit 与值均为零。其权重为 $\alpha_{0,t,h} = (1 + Z_{t,h})^{-1}$ ，其中 $Z_{t,h} = \sum_{j \in I_{b,g(h)}} \exp(\ell_{tj,h})$ 。锚点库非空时，读取结果可分解为

$$e_{t,h} = (1 - \alpha_{0,t,h}) \bar{v}_{t,h}, \quad \bar{v}_{t,h} = \sum_{j \in I_{b,g(h)}} \frac{\exp(\ell_{tj,h})}{Z_{t,h}} v_{j,g(h)}. \quad (15)$$

查询同时决定检索内容 $\bar{v}_{t,h}$ 和分配给记忆的总注意力权重。读取结果位于已存储值与零向量的凸包内；锚点库为空时，结果严格为零。

3.5 由查询调节记忆输出

输出门控逐头作用于检索结果：

$$g_{t,h} = \tanh(s_h) 2\sigma\left(\frac{\bar{q}_{t,h}^\top b_h}{\sqrt{d}} + c_h\right), \quad (16)$$

$$\text{RMA}(h_t, \mathcal{M}_{b(t)}) = W_O \text{concat}_h(g_{t,h}e_{t,h}). \quad (17)$$

其中， $\bar{q}$ 是旋转编码前的归一化查询； $s_h, c_h \in \mathbb{R}$ 、 $b_h \in \mathbb{R}^d$ 与 $W_O \in \mathbb{R}^{D \times 16d}$ 均为可学习参数。门控系数取值于 $(-2, 2)$ ，其符号由可学习尺度 $\tanh(s_h)$ 决定，幅度由当前查询控制的 sigmoid 调节。

准入选择决定哪些记录可用，检索决定读取什么内容及分配多少注意力，有符号门控再控制这些信息对残差流的贡献。

与相关设计的比较。 其他设计也会在压缩历史或局部窗口之外保留少量键值记录。AnLLM 通过训练将每段文本汇总为一个锚点 token，只保留这些 token 的键和值 [29]，但锚点数量仍随序列增长。HOLA 为线性注意力模型增加有界循环键值缓存，按固定、无参数的优先级排序 [30]。LTE 则学习逐头评分，在 token 离开滑动窗口时决定是否保留 [31]。RMA 使用独立写入器生成专门的记忆记录，而非复制已有的键和值。因果评分器每 256 个 token 按组执行一次 top- K 选择，将记录提交到固定容量的锚点库。检索采用独立归一化和门控，与局部注意力共同支持转换后模型的推理。

4 上下文变长，历史内存为何不增长？

循环状态、局部窗口和锚点库的容量都是固定的，因此历史存储量和每个 token 对历史的访问量都有上界。

4.1 内存不随序列长度增长

设提示词长度为 P 、生成长度为 N ，固定模型维度、层数、精度、局部窗口 W 、记忆容量 K 和块大小 C 。以定长张量元素和元数据字为存储单位，完整持久历史状态的分配量为

$$S_{\text{history}} = S_{\text{GDN}} + S_{\text{SWA}} + S_{\text{committed}} + S_{\text{candidates}} + S_{\text{metadata}}. \quad (18)$$

循环矩阵、局部环形缓存、已提交锚点库、待提交候选、分数和索引均有固定容量。因此，对于单条序列，历史状态存储相对于 $P + N$ 的复杂度为 $O(1)$ 。

$$S_{\text{committed}} = 2 \times 8 \times 4 \times 1024 \times 256 \times 2 \text{ bytes} = 32 \text{ MiB}, \quad (19)$$

$$S_{\text{candidates}} = 2 \times 8 \times 4 \times 256 \times 256 \times 2 \text{ bytes} = 8 \text{ MiB}. \quad (20)$$

式中各因子依次对应键和值、模块数、KV 组数、记录数、头维度以及每个 BF16 元素占用的两个字节。RMA 的 KV 总计为**每序列 40 MiB**。覆盖 1,024 个位置的窗口最多再占用 32 MiB，因此**每条序列的注意力 KV 合计为 72 MiB**。在相同的八个注意力层、四个 KV 头、头维度与 BF16 精度下，256K token 的全注意力缓存需要 $32 \text{ KiB} \times 262,144 = 8 \text{ GiB}$ ，注意力 KV 因此减少 $1 - 72/8192 \approx 99.1\%$ 。表 3 进一步列出循环状态及辅助历史张量。

表 3 每条序列的持久历史状态。各项容量固定，不随上下文长度增长。

组成部分	存储类型	每序列 MiB	占总量比例
注意力 KV		71.97	73.6%
局部窗口（此前 1,023 个位置）	BF16	31.97	32.7%
RMA	BF16	40.00	40.9%
已提交锚点		32.00	32.7%
待提交候选记录		8.00	8.2%
循环状态		25.50	26.1%
循环矩阵	FP16	24.00	24.6%
卷积状态	BF16	1.50	1.5%
元数据（分数、位置、有效标记、计数器）		0.29	0.3%
合计		97.76	100.0%

4.2 每个 token 的计算量

每生成一个 token，模型更新固定大小的循环状态，每组最多读取 W 个局部位置和 K 个锚点。投影与前馈运算的维度固定，非注意力部分的成本记为 c_{fixed} 。块提交最多从 $K + C$ 条记录中进行选择。以排序复杂度作为选择过程的上界，可得

$$T_{\text{decode}}(N) = O(Nc_{\text{fixed}}) + O(N(W + K)) + O\left(\left\lceil \frac{N}{C} \right\rceil (K + C) \log(K + C)\right) = O(N). \quad (21)$$

配合固定大小分块的流式预填充，**完整序列处理的总计算量为 $O(P + N)$** 。全注意力在解码阶段需要 $O(PN + N^2)$ 的注意力计算和 $O(P + N)$ 的历史存储。潜在表示压缩、跨 token 压缩、稀疏读取和混合层比例能够降低常数，但历史存储仍随 $P + N$ 线性增长；EdgeRock 消除了这种长度依赖。

4.3 推理缓存与请求并发

历史状态存储随并发请求数 B 增加，不随请求长度增加。运行时利用状态形状固定的特点，通过 CUDA Graphs、打包投影、融合更新和物理滑动窗口缓存优化执行，并为每条请求分配独立的历史槽位。相同提示词的请求可以共享预填充计算，但各条续写仍保留自己的状态。

分块预填充或恢复请求时，循环状态、局部 KV、已提交记录、待提交候选和位置必须一并传递；续写语义见附录 E。

5 评测设计

我们在 29 项文本基准上比较 28 个模型，并评估视觉推理、远距离记忆和长序列推理效率。完整文本成绩见附录 B 和 C。

5.1 比较目标

评测分为五组。与 16 个有界状态模型（含三个 14B 级模型）比较，考察恒定内存模型的能力水平；与源模型 Qwen3.5-4B 比较，衡量转换后的能力变化；与 MiniCPM5-2B、LFM2.5-2.6B 等小尺寸历史增长型模型比较，考察同规模模型间的能力差异。记忆实验通过启停 RMA 和成对修改远距离证据，验证模型是否依赖锚点作答。系统实验以四个小尺寸模型为参照，比较解码速率、批量吞吐量和显存占用。

有界状态对照组。 该组涵盖循环模型、状态空间模型、循环与局部注意力混合模型 [11, 32, 33, 34, 35, 36, 14, 37] 及窗口注意力模型。Phi-3 Medium 4K 使用 2,047 个 token 的滑动窗口 [38]。

5.2 指标与执行配置

多数任务报告准确率，DROP 报告 F1。各项成绩遵循基准规定的学科或子任务汇总方式；总体均值对 29 个主要指标等权平均，AIME 对 2024–2026 三届成绩等权平均。比较三个及以上模型的表格以粗体标出各指标最优值，并列最优一并标出；与 Qwen3.5-4B 的成对比较则报告能力保留率。

记忆与视觉评测使用参考实现的计算路径，运行时实验使用第 8 节的优化配置。采样、评分及源模型设置见附录 A。

6 恒定内存下，推理能力有多强？

EdgeRock-4B 将有界状态模型的推理能力推进到实用水平。在 17 个有界状态模型中，它在 29 项文本基准的 24 项上排名第一（附录 B）；AIME 成绩进入领先小尺寸历史增长型模型的区间（第 6.3 节）。有界状态模型的比较反映各架构及其训练方案的综合表现；与源模型 Qwen3.5-4B 的比较则衡量转换后的能力保留情况（第 6.2 节）。

6.1 明显领先 16 个恒定内存模型

EdgeRock-4B 在 29 项基准均值和竞赛数学两方面均领先于所评测的有界状态模型（图 3），对照包括参数规模更大的 GDN、RWKV 和 Mamba 模型 [35, 32, 36]。其 29 项基准等权均值比该指标上最强的基线 AHN-GDN Qwen2.5 14B 高出 **23.8 个百分点**。

在 AIME 2024–2026 上，EdgeRock-4B 的平均一次通过率（pass@1）为 **82.9%**；同组中该项最强的对照模型 RWKV-7 G1j 13.3B 为 18.8%（图 3）。

6.2 保留源模型 96% 的推理能力

EdgeRock-4B 基于 Qwen3.5-4B 的预训练权重进行转换。将全注意力替换为局部注意力与固定容量锚点库后，EdgeRock-4B 的 AIME 2024–2026 等权平均一次通过率保持在 Qwen3.5-4B 的 **96.3%**，MATH-500 达到 **97.5%**（表 4）。

表 4 竞赛数学与 MATH-500（平均一次通过率，%）。保留率为 EdgeRock-4B 得分占源模型 Qwen3.5-4B 得分的比例。AIME 对 2024–2026 三届等权平均。

基准	EdgeRock-4B	Qwen3.5-4B [5]	保留率 (%)
AIME 2024 [39]	86.7	89.6	96.8
AIME 2025 [39]	77.4	80.8	95.8
AIME 2026 [39]	84.6	87.9	96.2
HMMT Feb 2025 [40]	71.8	74.6	96.2
HMMT Nov 2025 [40]	69.8	78.3	89.1
HMMT Feb 2026 [40]	63.6	65.9	96.5
MATH-500 [41, 42]	97.5	98.0	99.5
AIME 三年均值	82.9	86.1	96.3

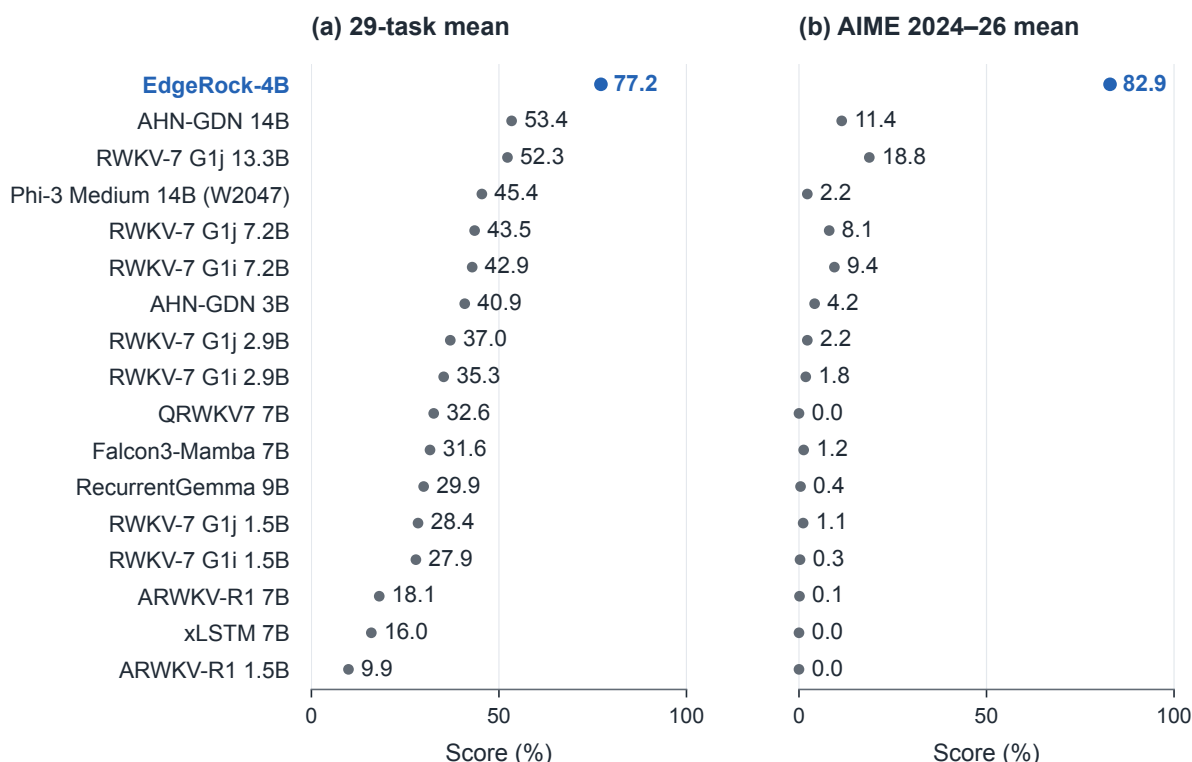


图3 有界状态模型的能力比较。(a) 29项文本基准的等权均值。(b) AIME 2024–2026 的等权均值。两个子图均按 29 项基准均值排列模型。在覆盖 1.5B–14B 的 17 个模型中，EdgeRock-4B 在两项汇总指标上均居首位。W2047 表示 Phi-3 Medium 配置的注意力窗口，长度为 2,047 个 token。

在 29 项文本基准中，EdgeRock-4B 有八项超过 Qwen3.5-4B，包括 MMLU、MMLU-Pro、MMLU-Redux、SuperGPQA、HumanEval 和 MBPP（图 4）。

6.3 与领先小尺寸模型水平相当

EdgeRock-4B 在 AIME 2024–2026 上的 82.9% 介于 Granite4.2-3B 的 82.2% 与 Qwen3.5-4B 的 86.1% 之间；MiniCPM5-2B 为 87.8%。其余对照模型中，Nemotron-3-Nano-4B 和 LFM2.5-2.6B 分别为 57.1% 和 56.2%；Gemma4-E4B 和 Phi-4-mini-reasoning 分别为 40.1% 和 38.9%（表 5；附录 C）。EdgeRock-4B 的 29 项文本基准等权均值为 77.2%，MiniCPM5-2B 为 76.7%（附录 C）。在侧重知识推理的 MMLU-Pro 和 SuperGPQA 上，EdgeRock-4B 相对 MiniCPM5-2B 的优势更大（表 5）。

与 LFM2.5-2.6B 相比，EdgeRock-4B 在 29 项基准中的 28 项上领先（附录 C）。

表 5 小尺寸模型能力比较（%）。EdgeRock 的持久历史状态固定；分隔线下方为历史增长型模型。AIME 对 2024–2026 三届取平均，粗体表示各列最高值。

模型	AIME	MMLU Pro	Super GPQA	GPQA Diamond	LCB v6	IFEval
EdgeRock-4B	82.9	79.8	60.0	64.6	61.6	88.0
Qwen3.5-4B [5]	86.1	77.7	52.8	77.3	75.7	89.6
Gemma4-E4B-it [43]	40.1	69.0	41.1	55.6	69.1	84.3
Nemotron-3-Nano-4B [44]	57.1	56.8	37.4	43.4	66.4	89.5
Phi-4-mini-reasoning [45]	38.9	38.5	32.2	48.0	35.3	45.7
MiniCPM5-2B [46]	87.8	65.3	40.9	67.7	83.3	86.3
LFM2.5-2.6B [47]	56.2	59.9	27.1	58.2	59.7	93.3

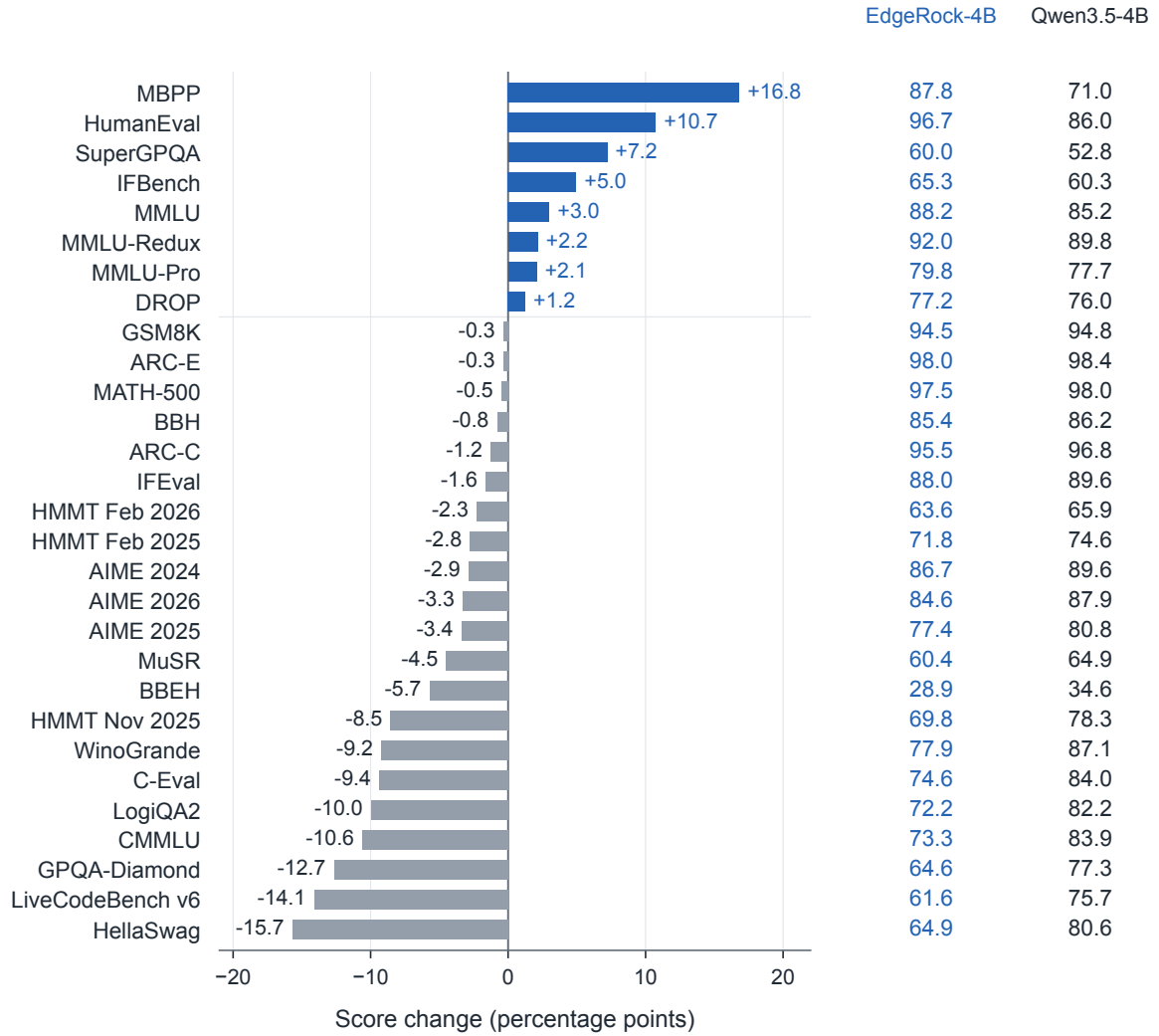


图 4 架构转换后的能力变化。条形表示 EdgeRock-4B 相对源骨干 Qwen3.5-4B 的得分差，单位为百分点，按差值从大到小排列。蓝色突出八项提升；右侧对齐的两列给出各模型的绝对得分（%）。

6.4 恒定内存下保留视觉能力

EdgeRock-4B 将恒定内存设计延伸到文本与图像推理。六项视觉基准覆盖感知、图表与示意图理解及视觉问答，EdgeRock-4B 的答案级准确率等权均值为 **84.9%**；Qwen3.5-4B 为 89.1%，对应能力保留率为 95.3%（表 6）。

表 6 视觉任务的答案级准确率（%），包括 MME。保留率为 EdgeRock-4B 得分占 Qwen3.5-4B 得分的比例。标准 MME 积分见附录 A。

基准	EdgeRock-4B	Qwen3.5-4B	保留率（%）
MMBench EN [48]	88.3	90.1	98.0
MMBench CN [48]	89.0	91.4	97.4
POPE [49]	85.4	88.8	96.2
MME [50]	83.0	86.8	95.6
ChartQA [51]	83.9	90.3	92.9
AI2D [52]	80.0	87.0	92.0
等权均值	84.9	89.1	95.3

7 RMA 能否支撑远距离推理？

RMA 使 EdgeRock-4B 能够利用局部窗口之外的证据。记忆测试集包含 256 个上下文，每个上下文有四个查询，输入长度为 16K 至 130,000 个 token。任务覆盖检索、状态更新、跨文档信息组合与变量绑定。启用 RMA 时，查询准确率为 **94.1%**；关闭后为 **0.4%**（图 5）。记忆能力在早期适配中形成，并在后训练中保留：查询准确率从 RMA 适配（M0）后的 79.4% 升至联合适配（R）后的 96.1%，最终模型保持在 94.1%（图 5a）。

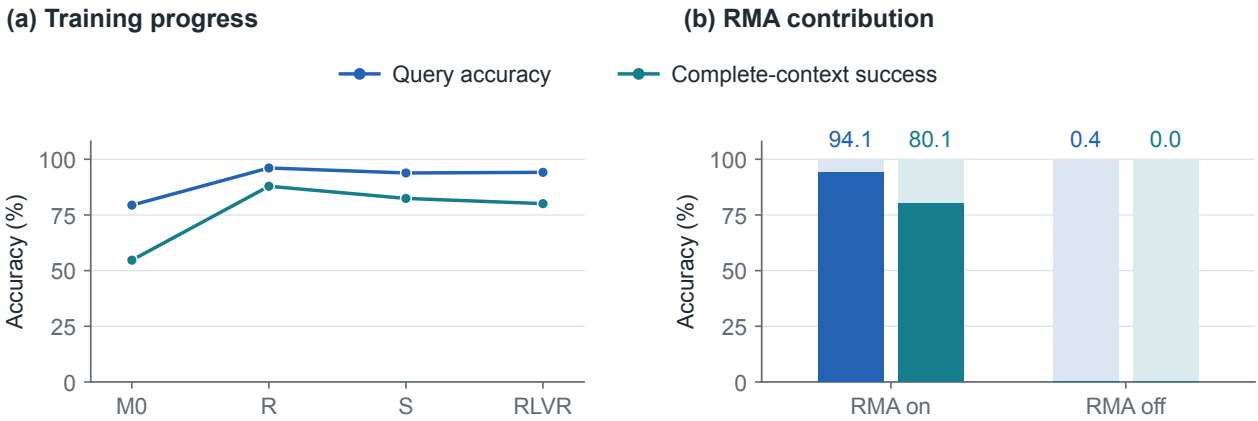


图 5 远距离记忆推理。(a) RMA 适配（M0）、联合适配（R）、监督微调（S）和可验证奖励强化学习（RLVR）后的查询准确率与完整上下文成功率。完整上下文成功要求四个答案全部正确。(b) 最终模型启用或关闭全部八个 RMA 模块的结果；两种设置都保留循环计算与局部注意力。启用 RMA 后，在检索、状态更新、跨文档信息组合和变量绑定任务中，查询准确率为 94.1%，完整上下文成功率为 80.1%。

7.1 没有 RMA，远距离回忆失效

推理时关闭全部八个 RMA 模块，循环骨干与局部注意力照常工作。1,024 个查询中的正确答案由 964 个降至 4 个；256 个上下文中，四个查询全部答对的数量由 205 个降至 0 个（图 5）。这表明，训练后的模型通过 RMA 读取作答所需的远距离信息。

7.2 RMA 在各类记忆任务中的表现

分任务结果。 EdgeRock-4B 在精确检索、状态更新解释和跨文档信息组合上的完整上下文成功率分别为 90.6%、87.5% 和 89.1%（表 7）。

成对证据干预。 修改远距离证据后，在答案应随之改变的 256 对样本中，EdgeRock-4B 同时答对两个版本的比例为 89.8%；在答案应保持不变的 256 对样本中，该比例为 96.5%（表 7）。

表 7 记忆机制诊断。上半部分按任务统计完整上下文成功率，要求同一上下文的四个答案全部正确，每项任务有 64 个上下文。下半部分成对修改远处证据，每类 256 对，要求两个答案均正确；对于答案应保持不变的样本对，未发生答案变化。

	正确数 / 总数	比例 (%)
各任务的完整上下文成功率		
精确检索	58 / 64	90.6
状态更新解释	56 / 64	87.5
跨文档信息组合	57 / 64	89.1
变量绑定	34 / 64	53.1
全部任务	205 / 256	80.1
远处证据的成对修改		
答案应改变	230 / 256	89.8
答案应保持不变	247 / 256	96.5

8 固定历史内存能带来哪些部署收益？

固定历史内存提高了推理服务效率。在单张 Hopper 80GB GPU 上，EdgeRock-4B 单请求生成 128K 输出的速率为 **233.7 tokens/s**。批量推理可同时运行 **512** 条长生成序列，端到端吞吐量平均为 Qwen3.5-4B 的 **4.94** 倍。收益来自固定的历史状态：每个 token 的计算量不随上下文增长，单卡也能同时运行更多序列；CUDA Graphs、融合 BF16 内核等实现细节见附录 D。

8.1 解码平稳，吞吐量更高

单请求解码。在 128K 输入、128K 输出和批大小为 1 的条件下，EdgeRock-4B 持续达到 **233.7 tokens/s**，为同一工作负载下 Qwen3.5-4B 的 **1.54** 倍。整个生成过程中的解码速率基本保持平稳（图 6a）。输入 254K、输出 2K（总长同为 256K）时，EdgeRock-4B 为 **226.6 tokens/s**，Qwen3.5-4B 为 133.9，前者为后者的 **1.69** 倍。

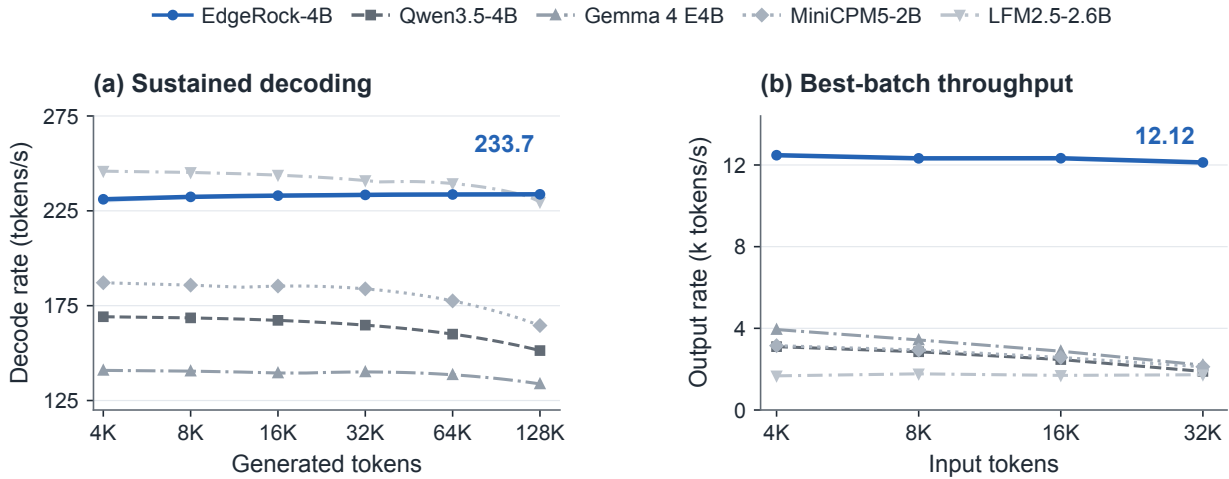


图 6 长上下文推理。EdgeRock-4B 与四个小尺寸模型均在单张 Hopper 80GB GPU 上使用 BF16 权重和 KV。(a) 128K 输入后生成 128K 输出、批大小为 1 时的累计解码速率，EdgeRock 在完整输出上的速率为 233.7 tokens/s。(b) 重复提示词、4K–32K 输入和 32K 输出下的输出吞吐量。仅统计输出 token，耗时包含预填充。各模型在每个输入长度下采用最佳实测批大小（附录表 19）；EdgeRock 在四种工作负载上均领先。

长生成吞吐量。长生成测试将输出固定为 32K，输入分别取 4K、8K、16K 和 32K。各运行时采用最佳实测批大小，EdgeRock-4B 的端到端吞吐量平均为 Qwen3.5-4B 的 **4.94** 倍、LFM2.5-2.6B 的 **7.18** 倍（附录表 19）。固定历史容量使更多请求能够同时运行。在 32K 输入、32K 输出下，Qwen3.5-4B 每条序列的注意力 KV 在生成结束时达到 2 GiB，每个 token 占 32 KiB。单张 80GB GPU 上并发几十条序列就会用满缓存，更多请求只能等待。EdgeRock 批量运行时的持久历史状态固定为每序列 97.76 MiB，因此 512 条序列可以一直并发运行到输出结束。

8.2 单卡并发扩展至 512 条序列

输入、输出各取 4K、8K、16K、32K，共 16 种组合；在单张 Hopper 80GB 上，EdgeRock-4B 在每种组合下都能并发运行 **512** 条序列。批大小为 512 时，加权解码吞吐量为 **12,497 tokens/s**，即每条活跃序列平均 **24.4 tokens/s**（图 7）。

8.3 短上下文下同样高效

在 4K 输入、2K 输出下，不计预填充时间，EdgeRock-4B 的解码速率在批大小为 1 时为 **229.6 tokens/s**，批大小为 512 时为 **12,481 tokens/s**。批大小为 1 时，张量显存占用峰值为 **14.27 GiB**（附录表 21）。

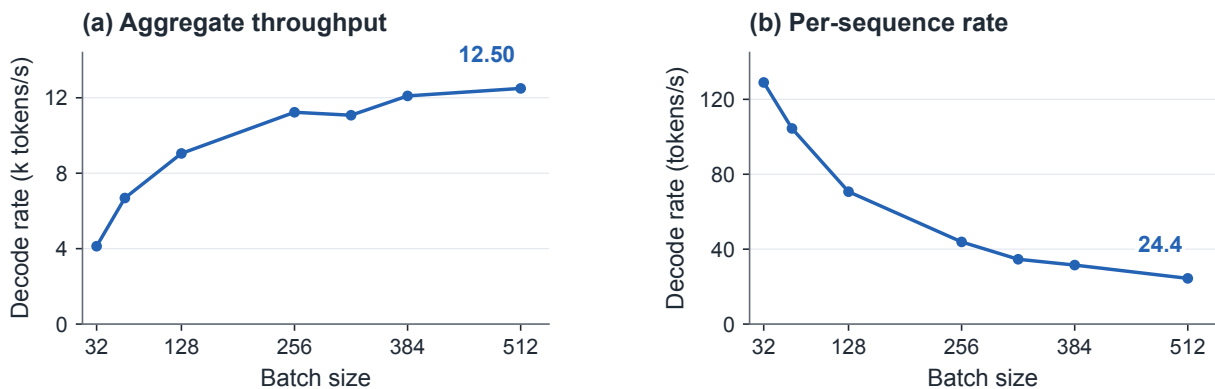


图 7 并发扩展。EdgeRock-4B 在单张 Hopper 80GB GPU 上运行 32–512 条活跃序列，权重与 KV 为 BF16。速率由 4K–32K 输入输出长度全部 16 种组合的解码 token 总数除以解码总耗时得到。(a) 总吞吐量。(b) 每条活跃序列的平均速率。

8.4 推理前即可确定内存需求

批量运行时，无论 EdgeRock-4B 推理 4K 还是 256K 个 token，每条序列的历史状态均为 **97.76 MiB**（表 3）。其中，**72 MiB** 的注意力 KV 比 Qwen3.5-4B 在 256K 长度下的缓存**减少 99.1%**（图 1b）。在单张 Hopper 80GB GPU 上，单条 128K 输入、128K 输出请求的张量显存占用峰值为 **16.02 GiB**；512 条序列并发时，全部 16 种工作负载中的最大值为 **61.26 GiB**（图 8）。

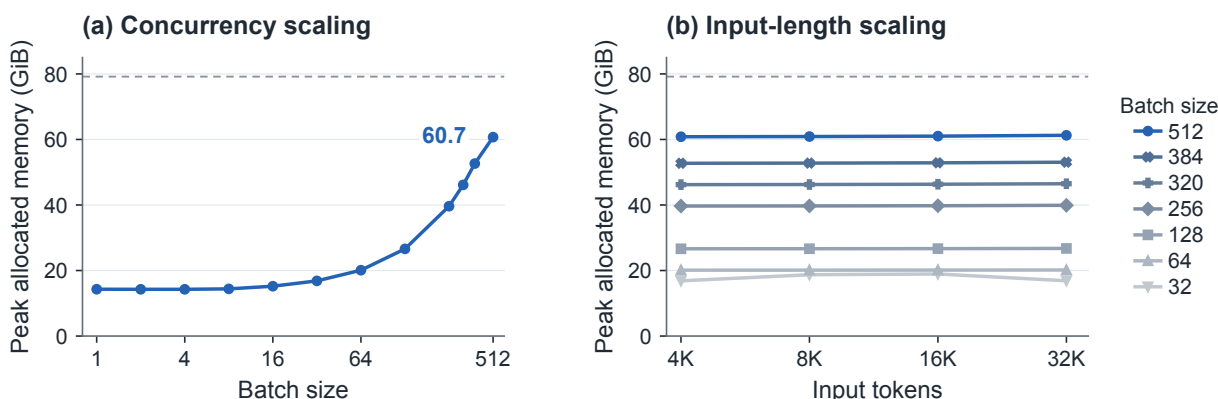


图 8 GPU 显存占用。EdgeRock-4B 在单张 Hopper 80GB GPU 上的张量显存占用峰值，包含权重、历史状态、缓冲区与工作区；虚线表示 GPU 容量。(a) 4K 输入、2K 输出下，内存随批大小的变化。(b) 输出固定为 32K 时，内存随输入长度的变化；各批大小下的分配量均基本保持不变。

9 结论

EdgeRock 表明，强大推理并不要求内存随 token 持续增长。一旦模型携带的历史状态固定下来，其存储需求便不再由输入长度决定，而由模型本身决定：设计时即可确定容量，推理过程中始终可以依此安排资源。

这让围绕推理模型构建系统有了新的可能。机器人可以让感知与推理在同一个循环中持续运行，历史内存需求不会随任务推进而增加。专用芯片可以把模型的全部历史状态放进片上存储，因为它不再随上下文长度变化。智能体的历史内存不再随工具调用次数增长。推理服务可以按请求数规划容量，而不必猜测每条请求最终会运行多久。

我们正在准备其他参数规模的 EdgeRock 模型，将 RMA 扩展到更大的骨干，检验恒定内存能否支撑更强的推理能力。这一转换方法也有望用于其他混合架构，让已有模型无需从头预训练即可采用固定历史状态。RMA 还可延伸到图像、音频和传感器数据流，这些场景尤其容易受到历史增长的制约。每个锚点都记录了来源位置，因此可以直接查看模型选择记住了什么。

参考文献

- [1] A. Vaswani et al. Attention Is All You Need. NeurIPS, 2017. arXiv:1706.03762.
- [2] W. Kwon et al. Efficient Memory Management for Large Language Model Serving with PagedAttention. SOSP, 2023. arXiv:2309.06180.
- [3] L. Chai et al. Accelerating LLM and VLM Inference for Automotive and Robotics with NVIDIA TensorRT Edge-LLM. NVIDIA Technical Blog, January 2026. <https://developer.nvidia.com/blog/accelerating-llm-and-vlm-inference-for-automotive-and-robotics-with-nvidia-tensorrt-edge-llm/>.
- [4] DeepSeek-AI. DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. arXiv:2405.04434, 2024.
- [5] Qwen Team. Qwen3.5. Model cards: [Qwen3.5-2B](#), [Qwen3.5-4B](#), 2026.
- [6] Kimi Team. Kimi Linear: An Expressive, Efficient Attention Architecture. arXiv:2510.26692, 2025.
- [7] DeepSeek-AI. DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models. arXiv:2512.02556, 2025.
- [8] DeepSeek-AI. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. arXiv:2606.19348, 2026.
- [9] Z. Liu et al. KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. ICML, 2024. arXiv:2402.02750.
- [10] A. Gu and T. Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. COLM, 2024. arXiv:2312.00752.
- [11] B. Peng et al. RWKV-7 “Goose” with Expressive Dynamic State Evolution. arXiv:2503.14456, 2025.
- [12] M. Beck et al. xLSTM: Extended Long Short-Term Memory. NeurIPS, 2024. arXiv:2405.04517.
- [13] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The Long-Document Transformer. arXiv:2004.05150, 2020.
- [14] A. Botev et al. RecurrentGemma: Moving Past Transformers for Efficient Open Language Models. arXiv:2404.07839, 2024.
- [15] L. Ren et al. Samba: Simple Hybrid State Space Models for Efficient Unlimited Context Language Modeling. ICLR, 2025. arXiv:2406.07522.
- [16] G. Xiao et al. Efficient Streaming Language Models with Attention Sinks. ICLR, 2024. arXiv:2309.17453.
- [17] Z. Zhang et al. H₂O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. NeurIPS, 2023. arXiv:2306.14048.
- [18] Y. Li et al. SnapKV: LLM Knows What You Are Looking for Before Generation. NeurIPS, 2024. arXiv:2404.14469.
- [19] S. Jelassi et al. Repeat After Me: Transformers are Better than State Space Models at Copying. ICML, 2024. arXiv:2402.01032.
- [20] S. Yang, J. Kautz, and A. Hatamizadeh. Gated Delta Networks: Improving Mamba2 with Delta Rule. ICLR, 2025. arXiv:2412.06464.
- [21] N. Lambert et al. Tülu 3: Pushing Frontiers in Open Language Model Post-Training. arXiv:2411.15124, 2024.
- [22] D. Guo et al. DeepSeek-R1 Incentivizes Reasoning in LLMs through Reinforcement Learning. Nature, 645:633–638, 2025. arXiv:2501.12948.
- [23] N. Shazeer. GLU Variants Improve Transformer. arXiv:2002.05202, 2020.
- [24] Rock Matrix. EdgeRock inference and deployment code. GitHub, revision 49e4b0f8fd0, accessed 11 October 2026. github.com/rockmatrix-ai/EdgeRock.
- [25] Rock Matrix. EdgeRock-4B model configuration. Hugging Face, revision e10f2b181881, accessed 11 October 2026. huggingface.co/rockmatrix-ai/EdgeRock.
- [26] J. Ainslie et al. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. EMNLP, 2023. arXiv:2305.13245.
- [27] B. Zhang and R. Sennrich. Root Mean Square Layer Normalization. NeurIPS, 2019. arXiv:1910.07467.
- [28] J. Su et al. RoFormer: Enhanced Transformer with Rotary Position Embedding. Neurocomputing, 568:127063, 2024. arXiv:2104.09864.
- [29] J. Pang et al. Anchor-based Large Language Models. Findings of ACL, 2024. arXiv:2402.07616.
- [30] W. Cui. A Hippocampus for Linear Attention: An Exact Memory for What the Recurrent State Forgets. arXiv:2607.02303, 2026.
- [31] M. He and P. N. Garner. Alleviating Forgetfulness of Linear Attention by Hybrid Sparse Attention and Contextualized Learnable Token Eviction. ICLR, 2026. arXiv:2510.20787.
- [32] RWKV Project. RWKV-7 G1 models. Model card, 2026. huggingface.co/BlinkDL/rwkv7-g1.
- [33] D. Goldstein et al. RADLADS: Rapid Attention Distillation to Linear Attention Decoders at Scale. COLM, 2025. arXiv:2505.03005.
- [34] Y. Lin et al. ARWKV: Pretrain Is Not What We Need, an RNN-Attention-Based Language Model Born from Transformer. arXiv:2501.15570, 2025. Model cards: [ARWKV-R1-1.5B](#), [ARWKV-R1-7B](#).
- [35] Y. Fang et al. Artificial Hippocampus Networks for Efficient Long-Context Modeling. ICML, 2026. arXiv:2510.07318.
- [36] Technology Innovation Institute. Falcon3-Mamba-7B-Instruct. Model card, 2024. huggingface.co/tiiuae/Falcon3-Mamba-7B-Instruct.
- [37] M. Beck et al. xLSTM 7B: A Recurrent LLM for Fast and Efficient Inference. ICML, 2025. arXiv:2503.13427.
- [38] M. Abdin et al. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219, 2024. Phi-3-medium-4k-instruct configuration: [config.json](#).
- [39] Mathematical Association of America. American Invitational Mathematics Examination (AIME), 2024–2026. <https://maa.org/maa-invitational-competitions/>.

- [40] Harvard–MIT Mathematics Tournament. HMMT February 2025, November 2025, and February 2026. <https://www.hmmt.org/www/archive/results>.
- [41] D. Hendrycks et al. Measuring Mathematical Problem Solving with the MATH Dataset. NeurIPS Datasets and Benchmarks, 2021. arXiv:2103.03874.
- [42] H. Lightman et al. Let’s Verify Step by Step. ICLR, 2024. arXiv:2305.20050.
- [43] Gemma Team. Gemma 4 Technical Report. arXiv:2607.02770, 2026. Model cards: [E2B](#), [E4B](#).
- [44] NVIDIA. NVIDIA-Nemotron-3-Nano-4B. Model card, 2026. huggingface.co/nvidia/NVIDIA-Nemotron-3-Nano-4B-BF16.
- [45] H. Xu et al. Phi-4-Mini-Reasoning: Exploring the Limits of Small Reasoning Language Models in Math. arXiv:2504.21233, 2025.
- [46] OpenBMB. MiniCPM5-2B. Model card, 2026. huggingface.co/openbmb/MiniCPM5-2B.
- [47] Liquid AI. LFM2 Technical Report. arXiv:2511.23404, 2025. LFM2.5-2.6B configuration, revision a4e00e83c097: [config.json](#).
- [48] Y. Liu et al. MMBench: Is Your Multi-modal Model an All-around Player? ECCV, 2024. arXiv:2307.06281.
- [49] Y. Li et al. Evaluating Object Hallucination in Large Vision-Language Models. EMNLP, 2023. arXiv:2305.10355.
- [50] C. Fu et al. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. NeurIPS Datasets and Benchmarks, 2025. arXiv:2306.13394.
- [51] A. Masry et al. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. Findings of ACL, 2022. arXiv:2203.10244.
- [52] A. Kembhavi et al. A Diagram Is Worth a Dozen Images. ECCV, 2016. arXiv:1603.07396.
- [53] K. Cobbe et al. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168, 2021.
- [54] D. Hendrycks et al. Measuring Massive Multitask Language Understanding. ICLR, 2021. arXiv:2009.03300.
- [55] Y. Wang et al. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. NeurIPS Datasets and Benchmarks, 2024. arXiv:2406.01574.
- [56] A. P. Gema et al. Are We Done with MMLU? NAACL, 2025. arXiv:2406.04127.
- [57] H. Li et al. CMMLU: Measuring Massive Multitask Language Understanding in Chinese. Findings of ACL, 2024. arXiv:2306.09212.
- [58] Y. Huang et al. C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models. NeurIPS Datasets and Benchmarks, 2023. arXiv:2305.08322.
- [59] D. Rein et al. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. COLM, 2024. arXiv:2311.12022.
- [60] X. Du et al. SuperGPQA: Scaling LLM Evaluation across 285 Graduate Disciplines. NeurIPS Datasets and Benchmarks, 2025. arXiv:2502.14739.
- [61] K. Sakaguchi et al. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. AAAI, 2020. arXiv:1907.10641.
- [62] R. Zellers et al. HellaSwag: Can a Machine Really Finish Your Sentence? ACL, 2019. arXiv:1905.07830.
- [63] P. Clark et al. Think You Have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. arXiv:1803.05457, 2018.
- [64] M. Suzgun et al. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. Findings of ACL, 2023. arXiv:2210.09261.
- [65] M. Kazemi et al. BIG-Bench Extra Hard. ACL, 2025. arXiv:2502.19187.
- [66] M. Chen et al. Evaluating Large Language Models Trained on Code. arXiv:2107.03374, 2021.
- [67] J. Austin et al. Program Synthesis with Large Language Models. arXiv:2108.07732, 2021.
- [68] N. Jain et al. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. ICLR, 2025. arXiv:2403.07974. Release v6: <https://github.com/LiveCodeBench/LiveCodeBench>.
- [69] J. Zhou et al. Instruction-Following Evaluation for Large Language Models. arXiv:2311.07911, 2023.
- [70] V. Pyatkin et al. Generalizing Verifiable Instruction Following. NeurIPS Datasets and Benchmarks, 2025. arXiv:2507.02833.
- [71] D. Dua et al. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. NAACL, 2019. arXiv:1903.00161.
- [72] Z. Sprague et al. MuSR: Testing the Limits of Chain-of-Thought with Multistep Soft Reasoning. ICLR, 2024. arXiv:2310.16049.
- [73] H. Liu et al. LogiQA 2.0: An Improved Dataset for Logical Reasoning in Natural Language Understanding. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2023. Dataset: <https://github.com/csitfun/LogiQA2.0>.
- [74] IBM. Granite 4.2. Model documentation, 2026. <https://www.ibm.com/granite/docs/models/granite4-2>. Model cards: [3B](#), [8B](#).
- [75] MiniCPM Team. MiniCPM4: Ultra-Efficient LLMs on End Devices. arXiv:2506.07900, 2025. MiniCPM4.1-8B model card: huggingface.co/openbmb/MiniCPM4.1-8B.
- [76] J. Shah et al. FlashAttention-3: Fast and Accurate Attention with Asynchrony and Low-precision. NeurIPS, 2024. arXiv:2407.08608.
- [77] T. Zadouri et al. FlashAttention-4: Algorithm and Kernel Pipelining Co-Design for Asymmetric Hardware Scaling. arXiv:2603.05451, 2026.
- [78] Z. Ye et al. FlashInfer: Efficient and Customizable Attention Engine for LLM Inference Serving. MLSys, 2025. arXiv:2501.01005.
- [79] L. Zheng et al. SGLang: Efficient Execution of Structured Language Model Programs. NeurIPS, 2024. arXiv:2312.07104.

A 评测与验证协议

文本评测与采样

29 项文本基准使用固定提示词、参考答案与可执行测试进行评测。

参数	EdgeRock-4B 文本评测设置
每题独立生成次数	少于 100 题时 $K_{\text{eval}} = 64$ ；100–10,000 题时为 16；超过 10,000 题时为 1
采样	温度 0.6；top- p 0.95；top- k 20；存在惩罚 0
种子 / 回答长度上限	20260923 / 131,072 个 token
停止条件	EOS、长度上限，或同一段 128–256 个 token 的机械循环原样重复八次

二元评分任务报告平均一次通过率 (pass@1)，并按各基准规定的学科或子任务方式汇总。不同模型、不同基准的重复次数可能不同。AIME 在每届内部汇总回答，再对 2024–2026 三届等权平均。

任务验证与汇总

验证器独立于参考答案，提取模型明确给出的答案，优先读取最终答案区域；语义评分也接受推理区域末尾已确定的答案。相互冲突或未确定的答案不得分。代码任务先采用宽松的程序提取规则，再运行可执行测试。IFEval 与 IFBench 使用严格约束检查器，语言检测器种子为 0。

BBH、BBEH 与 CMMLU 保留各自的子任务或学科宏平均，MMLU-Redux 使用修正后的选项。MuSR 对三个任务的准确率取平均。DROP 报告 token F1 和精确匹配率，其中 F1 计入 29 项基准均值。LogiQA2 使用所提供英文测试集的准确率。总体基准均值对各项指定的主要指标等权平均。

视觉与长程记忆评测

参数	视觉评测	长距离记忆诊断
生成方式	$K_{\text{eval}} = 1$ ；启用思考	贪心解码， $K_{\text{eval}} = 1$
计算路径 / 批大小	参考实现 / 128	参考实现 / 1
采样	温度 1.0；top- p 0.95；top- k 20；存在惩罚 1.5	贪心选择
种子	20260912	2026091707
回答长度上限	32,768 个 token	1,024 个 token
输入规模	65,536–4,194,304 像素	约 16K–130,000 个 token

六项视觉基准为 AI2D、ChartQA、MMBench EN/CN、MME 和 POPE，视觉基准均值对六项等权平均。按答案数加权的准确率、MME 类别积分与 MMBench 循环准确率分别统计。EdgeRock 的 MME 感知积分为 1,600.4，认知积分为 728.2，总分为 2,328.7。MMBench EN 循环准确率为 81.6%，六项基准按答案数加权的准确率为 85.5%。

记忆测试集包含 256 个上下文和 1,024 个查询。精确检索、状态更新解释、跨文档信息组合和变量绑定各有 64 个上下文，每个上下文四个查询。查询准确率逐答案计算；完整上下文成功要求四个答案全部正确。成对干预在保持周边结构不变的情况下修改远处证据，分别测试答案应改变与应保持不变的情形；成对准确率要求两个答案都正确。关闭 RMA 的干预保留已训练的骨干与局部窗口。

B 有界状态模型的完整基准结果

结果涵盖 EdgeRock-4B 与 16 个有界状态对照模型的 29 项基准。各任务采用其主要指标，数值范围为 0–100；粗体标出各列最高值，包括并列最高。历史增长型模型的结果见附录 C。

表 8 有界状态模型的数学成绩 (%)。MATH 指 MATH-500。

模型	GSM8K [53]	MATH [42]	AIME '24 [39]	AIME '25 [39]	AIME '26 [39]	HMMT Feb '25 [40]	HMMT Nov '25 [40]	HMMT Feb '26 [40]
EdgeRock-4B	94.5	97.5	86.7	77.4	84.6	71.8	69.8	63.6
ARWKV-R1 1.5B [34]	24.8	13.6	0.0	0.0	0.0	0.4	0.8	1.1
RWKV-7 G1i 1.5B [32]	60.3	28.2	0.0	0.4	0.4	1.7	5.0	0.0
RWKV-7 G1j 1.5B [32]	57.2	31.4	1.2	2.1	0.0	1.7	4.2	0.0
RWKV-7 G1i 2.9B [32]	71.7	43.8	1.7	1.7	2.1	1.7	5.0	0.8
RWKV-7 G1j 2.9B [32]	74.1	48.2	0.4	4.6	1.7	2.1	5.0	0.4
AHN-GDN Qwen2.5 3B [35]	83.8	66.2	7.1	2.1	3.3	2.9	6.7	4.9
ARWKV-R1 7B [34]	51.2	33.2	0.0	0.0	0.4	0.8	1.7	1.1
Falcon3-Mamba 7B Instruct [36]	61.4	41.8	1.7	0.8	1.2	0.8	5.4	2.7
QRWKV7 7B Instruct [33]	65.5	41.0	0.0	0.0	0.0	0.0	6.7	0.0
xLSTM 7B [37]	43.4	25.0	0.0	0.0	0.0	1.7	5.0	1.5
RWKV-7 G1i 7.2B [32]	85.1	68.6	8.8	11.7	7.9	2.1	5.0	3.4
RWKV-7 G1j 7.2B [32]	81.8	64.2	8.3	11.2	4.6	4.2	4.2	0.8
RecurrentGemma 9B IT [14]	51.7	22.8	0.4	0.8	0.0	1.2	5.4	1.9
RWKV-7 G1j 13.3B [32]	94.6	77.2	17.5	27.5	11.2	3.3	5.8	12.1
AHN-GDN Qwen2.5 14B [35]	93.6	80.0	13.8	10.0	10.4	7.5	10.0	4.9
Phi-3 Medium 4K Instruct [38]	85.3	49.0	3.3	3.3	0.0	6.7	2.5	3.0

表 9 有界状态模型的知识与科学成绩。Pro、Redux 分别指 MMLU-Pro、MMLU-Redux；GPQA 指 GPQA-Diamond；Super 指 SuperGPQA。

模型	MMLU [54]	Pro [55]	Redux [56]	CMMLU [57]	C-Eval [58]	GPQA [59]	Super [60]
EdgeRock-4B	88.2	79.8	92.0	73.3	74.6	64.6	60.0
ARWKV-R1 1.5B [34]	17.4	7.5	17.9	15.6	15.2	13.1	5.8
RWKV-7 G1i 1.5B [32]	49.7	13.3	55.4	33.6	33.0	26.8	17.7
RWKV-7 G1j 1.5B [32]	49.3	16.6	53.8	35.3	33.3	26.3	16.1
RWKV-7 G1i 2.9B [32]	63.6	37.8	63.4	36.1	32.5	26.8	21.6
RWKV-7 G1j 2.9B [32]	64.7	37.9	58.9	37.9	36.5	31.8	21.8
AHN-GDN Qwen2.5 3B [35]	62.7	42.8	65.7	62.4	62.9	31.3	23.0
ARWKV-R1 7B [34]	32.3	8.8	35.2	29.0	29.0	13.1	8.8
Falcon3-Mamba 7B Instruct [36]	57.7	26.1	64.8	33.4	35.2	20.7	12.2
QRWKV7 7B Instruct [33]	64.6	23.0	34.1	57.7	56.0	8.6	15.4
xLSTM 7B [37]	19.6	4.6	37.9	16.9	19.5	12.1	5.3
RWKV-7 G1i 7.2B [32]	69.4	44.9	73.5	49.8	51.6	30.3	25.7
RWKV-7 G1j 7.2B [32]	67.5	46.2	72.7	50.8	48.9	36.9	25.3
RecurrentGemma 9B IT [14]	55.5	13.4	55.0	43.4	41.7	24.7	19.5
RWKV-7 G1j 13.3B [32]	76.7	47.6	81.8	67.4	66.3	42.4	31.8
AHN-GDN Qwen2.5 14B [35]	77.2	63.5	83.5	75.0	74.2	40.9	35.5
Phi-3 Medium 4K Instruct [38]	76.3	54.0	81.4	53.3	51.3	26.8	25.6

表 10 有界状态模型的常识与推理成绩。Wino. 指 WinoGrande, Hella. 指 HellaSwag。

模型	Wino. [61]	Hella. [62]	ARC-C [63]	ARC-E [63]	BBH [64]	BBEH [65]
EdgeRock-4B	77.9	64.9	95.5	98.0	85.4	28.9
ARWKV-R1 1.5B [34]	31.4	7.4	19.6	22.3	9.8	1.6
RWKV-7 G1i 1.5B [32]	46.6	33.4	68.6	83.0	27.8	4.6
RWKV-7 G1j 1.5B [32]	49.3	24.6	66.2	80.8	31.4	5.0
RWKV-7 G1i 2.9B [32]	57.4	62.7	81.9	92.2	30.3	4.5
RWKV-7 G1j 2.9B [32]	58.1	65.9	83.3	91.6	39.5	6.5
AHN-GDN Qwen2.5 3B [35]	61.7	52.3	76.6	86.1	39.5	4.6
ARWKV-R1 7B [34]	34.4	10.0	47.4	56.8	16.6	2.4
Falcon3-Mamba 7B Instruct [36]	57.5	38.6	74.8	87.6	32.8	3.4
QRWKV7 7B Instruct [33]	61.8	48.0	86.3	95.4	37.4	1.9
xLSTM 7B [37]	28.3	7.9	25.1	28.2	12.8	3.3
RWKV-7 G1i 7.2B [32]	60.4	70.2	84.5	94.5	38.8	7.0
RWKV-7 G1j 7.2B [32]	59.4	71.7	83.7	94.5	41.4	7.7
RecurrentGemma 9B IT [14]	56.4	45.8	73.3	87.2	35.5	3.2
RWKV-7 G1j 13.3B [32]	72.1	77.4	91.8	96.3	48.7	7.7
AHN-GDN Qwen2.5 14B [35]	74.0	67.9	91.5	94.0	52.0	5.3
Phi-3 Medium 4K Instruct [38]	79.6	74.0	89.8	95.3	53.3	3.1

表 11 有界状态模型的代码与指令遵循成绩。LCB 指 LiveCodeBench。

模型	HumanEval [66]	MBPP [67]	LCB v6 [68]	IFEval [69]	IFBench [70]
EdgeRock-4B	96.7	87.8	61.6	88.0	65.3
ARWKV-R1 1.5B [34]	4.3	5.4	0.3	11.1	8.3
RWKV-7 G1i 1.5B [32]	42.1	36.6	7.2	37.0	14.7
RWKV-7 G1j 1.5B [32]	41.5	38.6	6.3	46.2	16.0
RWKV-7 G1i 2.9B [32]	58.5	49.2	11.6	43.1	17.0
RWKV-7 G1j 2.9B [32]	58.5	48.2	12.1	56.4	17.3
AHN-GDN Qwen2.5 3B [35]	67.7	49.0	14.6	58.6	26.3
ARWKV-R1 7B [34]	16.5	21.2	2.6	10.4	8.3
Falcon3-Mamba 7B Instruct [36]	34.1	39.2	9.3	64.7	18.7
QRWKV7 7B Instruct [33]	62.2	30.2	12.4	59.0	21.3
xLSTM 7B [37]	36.6	9.2	2.2	34.0	14.7
RWKV-7 G1i 7.2B [32]	65.2	59.0	18.8	53.4	17.7
RWKV-7 G1j 7.2B [32]	72.6	58.0	21.3	66.5	18.3
RecurrentGemma 9B IT [14]	37.2	19.0	7.9	39.7	12.7
RWKV-7 G1j 13.3B [32]	86.0	67.4	30.7	77.8	27.3
AHN-GDN Qwen2.5 14B [35]	75.6	68.6	32.9	77.4	33.3
Phi-3 Medium 4K Instruct [38]	73.8	59.4	19.1	61.4	22.0

表 12 有界状态模型的阅读与多步推理成绩。DROP 为 F1，MuSR 为任务宏平均准确率，LogiQA2 为准确率。

模型	DROP F1 [71]	MuSR [72]	LogiQA2 [73]
EdgeRock-4B	77.2	60.4	72.2
ARWKV-R1 1.5B [34]	5.6	8.6	17.5
RWKV-7 G1i 1.5B [32]	23.0	19.3	38.7
RWKV-7 G1j 1.5B [32]	24.9	29.1	36.1
RWKV-7 G1i 2.9B [32]	24.6	31.9	47.6
RWKV-7 G1j 2.9B [32]	30.7	34.5	45.4
AHN-GDN Qwen2.5 3B [35]	32.8	37.8	50.1
ARWKV-R1 7B [34]	11.1	15.4	26.8
Falcon3-Mamba 7B Instruct [36]	19.9	29.1	41.4
QRWKV7 7B Instruct [33]	21.2	2.6	33.2
xLSTM 7B [37]	29.0	19.3	20.0
RWKV-7 G1i 7.2B [32]	33.3	49.6	53.1
RWKV-7 G1j 7.2B [32]	34.2	48.6	56.4
RecurrentGemma 9B IT [14]	37.1	35.2	40.2
RWKV-7 G1j 13.3B [32]	51.5	51.4	66.6
AHN-GDN Qwen2.5 14B [35]	66.5	56.4	72.3
Phi-3 Medium 4K Instruct [38]	51.5	51.1	62.7

C 历史增长型模型的完整基准结果

本节在 29 项基准上比较 EdgeRock-4B 与十一个历史增长型模型。粗体标出各列最高值，包括并列最高。指标和采样规则见附录 A。

表 13 与历史增长型模型的数学能力比较 (%)。MATH 指 MATH-500。

模型	GSM8K [53]	MATH [42]	AIME '24 [39]	AIME '25 [39]	AIME '26 [39]	HMMT Feb '25 [40]	HMMT Nov '25 [40]	HMMT Feb '26 [40]
EdgeRock-4B	94.5	97.5	86.7	77.4	84.6	71.8	69.8	63.6
Qwen3.5-4B [5]	94.8	98.0	89.6	80.8	87.9	74.6	78.3	65.9
Gemma4-E2B-it [43]	90.1	78.8	38.3	33.8	33.8	21.7	17.5	20.1
MiniCPM5-2B [46]	93.6	97.0	87.5	85.4	90.4	76.7	74.6	63.3
LFM2.5-2.6B [47]	91.0	92.7	63.4	56.3	48.9	34.7	42.9	32.9
Qwen3.5-2B [5]	87.6	86.8	41.2	34.2	35.4	26.2	23.3	25.4
Granite4.2-3B [74]	93.8	97.8	85.0	78.8	82.9	75.8	72.5	59.5
Phi-4-mini-reasoning [45]	93.6	91.6	45.4	32.5	38.8	25.0	28.7	24.6
Gemma4-E4B-it [43]	91.3	88.2	42.1	40.0	38.3	25.4	32.9	28.8
Nemotron-3-Nano-4B [44]	92.4	94.8	60.8	56.7	53.8	40.0	47.1	36.4
Granite4.2-8B [74]	94.4	98.0	88.3	87.9	91.7	81.7	86.7	73.9
MiniCPM4.1-8B [75]	93.6	96.4	77.1	70.0	70.8	58.3	59.6	49.2

表 14 与历史增长型模型的知识与科学能力比较。Pro、Redux 分别指 MMLU-Pro、MMLU-Redux；GPQA 指 GPQA-Diamond；Super 指 SuperGPQA。

模型	MMLU [54]	Pro [55]	Redux [56]	CMMLU [57]	C-Eval [58]	GPQA [59]	Super [60]
EdgeRock-4B	88.2	79.8	92.0	73.3	74.6	64.6	60.0
Qwen3.5-4B [5]	85.2	77.7	89.8	83.9	84.0	77.3	52.8
Gemma4-E2B-it [43]	72.2	47.0	78.2	60.2	61.2	46.5	34.1
MiniCPM5-2B [46]	79.6	65.3	85.3	79.2	78.3	67.7	40.9
LFM2.5-2.6B [47]	75.3	59.9	81.7	57.2	59.5	58.2	27.1
Qwen3.5-2B [5]	72.5	62.8	79.5	68.2	68.7	54.5	39.3
Granite4.2-3B [74]	77.2	66.4	81.6	59.1	60.8	58.6	39.5
Phi-4-mini-reasoning [45]	77.5	38.5	78.3	45.3	42.4	48.0	32.2
Gemma4-E4B-it [43]	81.2	69.0	86.4	68.7	68.6	55.6	41.1
Nemotron-3-Nano-4B [44]	75.8	56.8	81.5	54.6	56.4	43.4	37.4
Granite4.2-8B [74]	78.6	72.0	86.3	55.1	58.4	61.6	45.7
MiniCPM4.1-8B [75]	82.7	67.0	87.8	80.2	79.0	51.5	40.6

表 15 与历史增长型模型的常识与推理能力比较。Wino. 指 WinoGrande，Hella. 指 HellaSwag。

模型	Wino. [61]	Hella. [62]	ARC-C [63]	ARC-E [63]	BBH [64]	BBEH [65]
EdgeRock-4B	77.9	64.9	95.5	98.0	85.4	28.9
Qwen3.5-4B [5]	87.1	80.6	96.8	98.4	86.2	34.6
Gemma4-E2B-it [43]	63.2	50.9	89.2	94.8	80.3	11.6
MiniCPM5-2B [46]	69.8	52.2	91.9	97.0	82.1	25.4
LFM2.5-2.6B [47]	67.9	40.7	91.6	96.5	76.7	13.1
Qwen3.5-2B [5]	71.5	59.3	91.4	95.5	79.1	14.9
Granite4.2-3B [74]	74.0	67.4	92.4	97.1	85.5	24.5
Phi-4-mini-reasoning [45]	75.8	62.6	90.2	94.4	72.5	12.9
Gemma4-E4B-it [43]	76.5	53.5	94.3	97.7	84.1	17.4
Nemotron-3-Nano-4B [44]	65.6	46.3	91.9	96.7	79.3	10.8
Granite4.2-8B [74]	76.2	63.1	90.9	94.6	88.9	31.5
MiniCPM4.1-8B [75]	78.2	49.8	93.4	97.4	86.6	14.0

表 16 与历史增长型模型的代码与指令遵循能力比较。LCB 指 LiveCodeBench。

模型	HumanEval [66]	MBPP [67]	LCB v6 [68]	IFEval [69]	IFBench [70]
EdgeRock-4B	96.7	87.8	61.6	88.0	65.3
Qwen3.5-4B [5]	86.0	71.0	75.7	89.6	60.3
Gemma4-E2B-it [43]	90.9	79.6	59.6	82.4	29.0
MiniCPM5-2B [46]	97.6	94.8	83.3	86.3	64.0
LFM2.5-2.6B [47]	92.9	73.6	59.7	93.3	56.2
Qwen3.5-2B [5]	54.9	35.4	25.0	81.5	43.7
Granite4.2-3B [74]	97.0	92.6	75.9	92.2	74.3
Phi-4-mini-reasoning [45]	79.9	64.0	35.3	45.7	14.3
Gemma4-E4B-it [43]	93.9	79.2	69.1	84.3	34.7
Nemotron-3-Nano-4B [44]	90.9	72.2	66.4	89.5	57.7
Granite4.2-8B [74]	96.3	95.4	85.6	94.6	77.0
MiniCPM4.1-8B [75]	98.2	92.2	70.4	73.4	23.7

表 17 与历史增长型模型的阅读与多步推理能力比较。DROP 为 F1，MuSR 为任务宏平均准确率，LogiQA2 为准确率。

模型	DROP F1 [71]	MuSR [72]	LogiQA2 [73]
EdgeRock-4B	77.2	60.4	72.2
Qwen3.5-4B [5]	76.0	64.9	82.2
Gemma4-E2B-it [43]	72.9	58.9	59.4
MiniCPM5-2B [46]	76.0	65.3	73.7
LFM2.5-2.6B [47]	72.3	60.4	63.2
Qwen3.5-2B [5]	73.7	61.3	66.3
Granite4.2-3B [74]	83.7	65.1	69.5
Phi-4-mini-reasoning [45]	50.6	56.7	58.5
Gemma4-E4B-it [43]	72.9	62.2	72.6
Nemotron-3-Nano-4B [44]	72.6	58.2	67.9
Granite4.2-8B [74]	76.9	62.4	73.8
MiniCPM4.1-8B [75]	78.3	64.5	75.7

D 运行时协议与内存核算

D.1 工作负载与硬件

所有测试使用单张 NVIDIA Hopper 80GB GPU，张量并行度为一。全部矩阵权重与 KV 均为 BF16。

本文 1K = 1,024 个 token。持续单请求测试的批大小为 1，输入和输出各为 128K。另一组单请求测试使用 254K 输入与 2K 输出，EdgeRock 和 Qwen 各独立运行三次；两组测试的总长度均为 262,144 个 token。

长生成吞吐量测试使用 4K、8K、16K、32K 输入和 32K 输出。EdgeRock 的完整测试矩阵将这四种输入长度与同样四种输出长度交叉组合，批大小取 32、64、128、256、320、384 和 512。另设 4K 输入、2K 输出测试，覆盖批大小 1、2、4、8、16、32、64、128、256、320、384 和 512。每个 EdgeRock 配置独立运行三次；未注明加权的速率均取中位数。

测试使用合成输入 token，忽略 EOS，在目标输入长度和批容量下先预热 256 个 token。EdgeRock 采用全词表自回归采样，温度为 1.0，top- k 为 20，top- p 为 0.95，存在惩罚为 1.5。

D.2 计时区间与汇总

解码速率统计每条请求首个输出 token 之后的生成过程，计时包含采样、状态更新、图管理和运行时调度。单请求曲线每生成 4K 个 token 报告一次累计速率。**端到端输出吞吐量**等于生成的输出 token 总数除以完整生成调用耗时，分母包含预填充时间，分子不计输入 token。两种指标均不包含模型加载、预热、分词和将 token 转为文本的耗时。

平均吞吐倍数由四种 32K 输出工作负载下的吞吐倍数等权平均得到；各运行时均采用最佳实测批大小。批量加权吞吐量汇总全部 16 种长度组合，每种组合运行三次。各次运行均统计首个输出 token 之后的 token 数，总数除以相应解码总耗时得到加权吞吐量，再除以活跃批大小得到单序列速率。阴影带表示各次运行的取值范围。内存结果采用各次运行的已分配与预留内存峰值。

D.3 运行时配置与提示复用

运行时通过打包投影以及融合归一化、旋转编码和记忆操作，减少数据搬运。批大小为 512 时，运行时让原始与打包后的 BF16 权重共享存储，节省 4.95 GiB。批大小为 1 时，运行时还把输出头处理、全词表采样与状态更新融合在一起。解码采用普通自回归方式，权重和历史状态均驻留 GPU。

EdgeRock 与 LFM 测试的批大小为 32、64、128、256、320、384 和 512；Qwen 与 MiniCPM 为 32 和 64；Gemma 为 64 和 128。批大小指提交的请求数。EdgeRock 保持目标并发数；vLLM 则在缓存预算内调度活跃请求。EdgeRock 批量矩阵中，每八条续写共享同一个提示词的预填充，之后各自保留独立的循环状态、局部注意力状态和锚点状态。短上下文测试中，最多八条请求共享一个提示词。

vLLM 对照模型使用 CUDA Graphs、分块预填充和 16K 的预填充 token 预算。Qwen、Gemma 与 MiniCPM 使用异步调度。Gemma、MiniCPM 与 LFM 的 GPU 内存利用率设置为 95%。合成单请求测试使用 256K 执行容

表 18 各模型在单张 Hopper 80GB GPU 上的运行时配置。

模型	历史状态精度	执行路径
EdgeRock-4B	BF16 KV; GDN 在批大小为 1 时以 FP32 存储, 其余为 FP16; 均以 FP32 更新	定制 CUDA Graph 引擎; 融合 BF16 内核; 直接索引状态与局部环形缓存
Qwen3.5-4B	BF16 KV; FP32 GDN 状态	vLLM [2] CUDA Graphs; FlashAttention-3 [76] (批大小为 1)、-4 [77] (批量); FlashInfer [78] GDN 预填充与融合解码
Gemma 4 E4B	BF16 KV	vLLM O3; FlashAttention-4; 仅语言模型部分
MiniCPM5-2B	BF16 KV	vLLM O3; FlashAttention-3
LFM2.5-2.6B	BF16 KV	vLLM 0.23.0; CUDA Graphs 与分块预填充

量, 权重与 RoPE 频率均不变。Gemma E4B 表示有效规模, 其完整检查点约为 8B 参数; MiniCPM 的检查点约为 2.52B 参数。

D.4 历史状态与显存分配

72 MiB 注意力 KV 是按配置计算的容量上界。解码时, 局部缓存保存此前 1,023 个位置, 当前 token 由解码器提供。表 3 列出的历史张量在批量运行时合计为每序列 97.76 MiB, 包括循环状态、卷积状态和锚点元数据。

张量显存占用峰值包括权重、历史状态、输入输出缓冲区、图存储及临时工作区。预留内存另含分配器缓存的内存块。批大小为 512 时, 原始与打包后的 BF16 投影共享存储, 省去权重副本, 历史容量不变。

D.5 吞吐量与显存实测

表 19–21 列出图 6–8 对应的测量值。

表 19 单张 Hopper 80GB 上生成 32K 输出时的输出吞吐量。仅统计输出 token, 耗时包含预填充。各列为输入长度, 单元格给出输出 tokens/s, 括号内为最佳实测批大小。粗体标出每列最高吞吐量。

模型	4K	8K	16K	32K
EdgeRock-4B	12,479 (512)	12,325 (512)	12,330 (512)	12,124 (512)
Qwen3.5-4B	3,103 (64)	2,853 (64)	2,472 (64)	1,889 (64)
Gemma 4 E4B	3,940 (128)	3,432 (128)	2,880 (128)	2,196 (64)
MiniCPM5-2B	3,152 (64)	2,938 (64)	2,575 (32)	2,117 (32)
LFM2.5-2.6B	1,673 (64)	1,771 (64)	1,695 (32)	1,727 (64)

表 20 单张 Hopper 80GB 上, 16 种输入输出组合 (4K–32K) 的加权解码速率与 GPU 内存峰值。

批大小	解码 tokens/s	每序列 tokens/s	已分配 GiB	预留 GiB
32	4,128	129.0	18.94	19.49
64	6,685	104.5	20.16	21.80
128	9,049	70.7	26.74	27.92
256	11,231	43.9	39.90	41.77
320	11,074	34.6	46.49	50.69
384	12,099	31.5	53.06	57.27
512	12,497	24.4	61.26	66.97

表 21 单张 Hopper 80GB 上，4K 输入、2K 输出时的解码吞吐量与 GPU 分配峰值。

批大小	解码 tokens/s	每序列 tokens/s	已分配 GiB
1	230	229.6	14.27
2	358	178.8	14.27
4	692	172.9	14.27
8	1,322	165.2	14.40
16	2,442	152.6	15.21
32	4,053	126.7	16.84
64	6,698	104.7	20.10
128	9,066	70.8	26.60
256	11,100	43.4	39.62
320	11,062	34.6	46.14
384	12,086	31.5	52.64
512	12,481	24.4	60.71

E 流式执行与状态归属

E.1 续写请求持有的状态

每条续写请求保存循环矩阵、局部 KV 环形缓存、已提交锚点库、待提交候选和进度元数据（表 22）。暂停或迁移请求槽位时，这些状态都必须保留。新锚点在下次块提交之后才可读取。

表 22 单条请求续写所需的逻辑状态。涉及 RMA 的容量按每个模块、每个 KV 组计算。

状态	内容	维护规则
循环状态	GDN 的历史表示	更新固定维度的循环矩阵。
局部历史	近期注意力 KV 与环形缓存位置	最多保留 $W = 1,024$ 个位置。
已提交锚点库	已选中的 KV 记录、准入分数和来源位置	每组最多保留 $K = 1,024$ 条记录。
待提交候选	当前块的记录、分数与已占用容量	提交前最多累积 $C = 256$ 个位置。
进度元数据	位置、块进度及与活跃请求的关联	维持续写语义及槽位归属的一致性。

模型权重由请求共享；历史状态保存在共享缓存池中的独立槽位。延长已有请求时复用其历史容量，接纳新请求时再增加一个槽位。

E.2 前向流程与因果可见性

预填充与解码都遵循算法 1，采用相同的逻辑提交边界。在每个包含 256 个 token 的块内，查询始终读取已提交锚点库，近期 token 则由局部注意力覆盖。边界提交完成后，被选中的候选记录从下一块起可用。

算法 1 单个局部注意力与 RMA 层的因果更新

输入： 隐藏输入 h_t 、因果索引 t 、旋转坐标 $\mathbf{p}_t$

输入： 局部状态 $\mathcal{L}$ 、已提交锚点库 $\mathcal{M}$ 、待提交候选 C

输入： 窗口 $W = 1,024$ 、锚点库容量 $K = 1,024$ 、候选块大小 $C = 256$

- 1: $u_t \leftarrow \text{Norm}_{\text{mem}}(h_t)$
 - 2: $m_t \leftarrow \text{ReadAndGate}(u_t, \mathbf{p}_t, \mathcal{M})$ ▷ 仅读取已提交锚点库
 - 3: $w_t \leftarrow \text{Writer}(u_t); (k_t, v_t) \leftarrow \text{MemoryKV}(w_t, \mathbf{p}_t)$
 - 4: $a_t \leftarrow \text{Admission}(u_t)$ ▷ 四个组的因果评分
 - 5: $(\ell_t, \mathcal{L}) \leftarrow \text{LocalAttentionStep}(h_t, \mathbf{p}_t, \mathcal{L}; W)$
 - 6: $z_t \leftarrow h_t + \ell_t + m_t; h_t^{\text{out}} \leftarrow z_t + \text{MLP}(\text{Norm}_{\text{ffn}}(z_t))$
 - 7: 将 (k_t, v_t, a_t, t) 加入 C
 - 8: 若 $|C| = C$ 则 ▷ 逻辑块边界，与预填充充分块无关
 - 9: 遍历 $g = 1, \dots, 4$ 执行
 - 10: $\mathcal{M}_g \leftarrow \text{TopK}_{a_g}(\mathcal{M}_g \cup C_g; K)$
 - 11: 结束循环
 - 12: $C \leftarrow \emptyset$ ▷ 更新后的库从下一块起可读
 - 13: 结束条件
 - 14: 返回 $h_t^{\text{out}}, \mathcal{L}, \mathcal{M}, C$
-

`ReadAndGate` 包括查询投影、带空项的已提交锚点库注意力、有符号门控和输出投影（第 3 节）。`MemoryKV` 对写入器输出执行键值投影、键归一化及旋转编码。局部注意力包含当前 token，最多保留 $W = 1,024$ 个位置。`Top-K` 同分时优先保留来源更早的记录。新序列从空局部状态、空锚点库和空候选缓冲区开始；跨越预填充充分块或暂停请求时，未完成的候选块仍须保留。

E.3 选择不变量与可见性

在块边界，每个 KV 组最多检查 1,280 条记录，保留其中最多 1,024 条，保留记录的键和值向量不变。由于分数在写入时确定、同分排序规则一致，式 10 中的历史 top- K 不变量始终成立。查询头共享所在组的锚点库，各自计算检索权重。局部注意力与记忆注意力分别归一化；空锚点库返回零。

E.4 续写、分块与生命周期

可移植后端默认按 1,024 个 token 对预填充充分块。分块限制工作集大小，同时保留局部、循环和锚点状态。RMA 的提交始终由候选块是否完成决定，与执行分块的边界无关。

从前缀恢复请求时，需要恢复表 22 中的全部状态，包括尚未提交的候选记录，使其能参与下一次提交。

原生适配器通过 GPU 上的请求槽位索引就地访问历史状态。SGLang [79] 的重置与复制操作作用于初始化重新分配的槽位，并保留续写请求的状态。

发布版本的位置配置。 RMA 对 256 个坐标中的 64 个使用部分 RoPE，基数为 10^7 ，采用默认缩放，最大执行位置设置为 262,144。多模态位置使用交错旋转分段 (11, 11, 10)；纯文本在三个轴上提供相同的累计 token 索引 [24, 25]。