20GB中的一百万个令牌：面向 Apple Silicon 上循环式 MoE 推理的无需校准 2 位 SRFT KV缓存压缩

Mohamed Amine Bergach
Illumina
mbergach@illumina.com

arXiv:2611.29092v1 [cs.PF] 21 Nov 2026

摘要—**在统一内存的边缘设备上进行长上下文推理，受 KV 缓存容量和内存带宽制约。在一台 32 GB Apple M6 Mac mini（26.8 GB Metal 工作集，150 GB/s DRAM）上，26.4B 的循环式混合专家模型 Inkling-2-Mini（48 次逻辑访问，1M 原生上下文）在原版 oMLX 中运行至 64K 词元时便耗尽内存（即使将其 40 次滑动窗口访问改为有界环形缓冲区，运行至 512K 时仍会耗尽内存）。我们发现，现代混合式无旋转位置编码架构的三项结构特性——有界滑动窗口、尾部恒为零的 1,024 词元带状相对偏置，以及单位 RMS 的值归一化——与正交归一化子采样随机傅里叶变换（SRFT）相结合，可对远端前缀中的 8 次全局访问实现**无需校准、无需尺度参数的 ***2.000*** **位值和 2.125 位键 Lloyd–Max 压缩（每坐标有效位数为 2.0625，** $7.76\times$ **小于 bf16）；远端前缀** $[0, N-1024)$**与精确的 bf16 局部窗口及学习型汇聚项合并，并通过在线 log-sum-exp 进行计算。自定义的 GQA打包Metal解码内核采用 256 项**`half4`** 查找表和 Block-8 延迟式** `exp2` **重缩放，达到 73.4 GB/s；在 256K 上以 17.09 令牌/秒运行 Inkling-2-Mini（**$3.36\times$ **比 bf16 更快），在 512K 上达到 13.81 令牌/秒，在 1,048,576 个词元上达到 9.52 令牌/秒，峰值内存为 19.49 GiB（KV 缓存为 4.37 GiB）。在** 4×4096 **个 WikiText-2 窗口中，3.06 位 SRFT 无损（**Δ**困惑度** $=-0.007$**），2.56 位 K3/V2 增加** $+0.024$**，2.06 位增加** $+0.130$**，同时在 8K、16K 和 32K 上实现 100%（3/3）的精确 7 位通行密钥检索。**索引词**—KV 缓存、2 位量化、SRFT、快速傅里叶变换、Apple Silicon、Metal、循环式Transformer、长上下文**

I. 引言

开放权重推理模型已迅速趋同于稀疏混合专家（MoE）骨干网络，配备 100 万 token 的上下文窗口，旨在适配 32GB 消费级统一内存设备。2026 年的代表性发布版本是 Thinking Machines Lab 的 Inkling-2-Mini [1]：拥有 264 亿个物理参数（采用混合 4/8 位 oQ4e 时占用 15.1GiB）、24 个物理层，其中间的 12 层会针对每个 token 循环三次（48 次逻辑访问）、滑动窗口（$W = 1024$）到全局注意力的比例为 5:1（40 次滑动访问、8 次全局访问），不使用旋转位置嵌入（NoPE），采用 1,024 token 的带状相对偏置、查询依赖型注意力汇聚项，以及单位均方根值归一化。

尽管 Inkling-2-Mini 的权重仅占用 15.1 GiB，但在 32 GB Apple M6 Mac mini 上以长上下文运行时，现有运行时会立即失败：

1) **无界滑动缓存：** 原版 oMLX [2] 为全部 48 次访问分配无界 KVCache，而不是为 40 次滑动访问轮换使用环形缓冲区；在 32K 时会消耗 6.05 GiB 的 KV，并在约 64K 时超出 M6 的 26.8GiB Metal 工作集（1M 则需要∼208 GiB）。

2) **bf16 全局键值墙：** 即使将 40 次滑动访问修复为有界 RotatingKVCache（已作为 oMLX PR #5512 贡献至上游），8 次全局访问仍需以 bf16 格式占用 32 KB/令牌——在 256K 时占用 8.2 GiB（解码速度降至 5.09 令牌/秒），在 1M 时占用 32.2 GiB（总计 47.3 GiB，在 32 GB 内存上物理上无法实现）。

3) **损失带宽的低比特路径：** 标准 MLX 和 llama.cpp 的量化 KV 路径要么物化稠密掩码/副本，要么在分组查询注意力（GQA）下针对每个查询头进行反量化；压缩 KV 的同时牺牲了解码速度。

本文展示了，将 Inkling-2-Mini 的三项架构特性与正交子采样随机傅里叶变换（SRFT） [3] 及自定义 GQA 打包 Metal 内核相结合，可在 32 GB Apple M6 上实现**完整的 1,048,576 令牌推理，仅占用 19.49 GiB，速度达 9.52 令牌/秒**。我们的贡献如下：

- **通过 SRFT 实现无尺度 2.000 位 Value 压缩：** 由于 Inkling-2 对 V 强制执行无需参数的单位 RMS 归一化，应用正交归一 SRFT 会将每个值向量旋转为各向同性的 $\mathcal{N}(0,\ 1)$ 类分布，在标量 Lloyd–Max 量化下无需零 每个 token 或每个块的尺度参数——以每坐标 2.0625 个有效比特（$7.76\times$ 压缩）实现 Value 压缩，无需离线校准，也没有浪费的 1 位 QJL 残差 [4]。

- **采用精确在线对数和指数合并的分割带状偏置注意力：** 将每次全局访问拆分为量化的远端前缀 $[0, N-1024)$（其中带状相对偏置恒为零）和精确的 bf16 近期窗口 $[N-1024, N)$，再加上学习型汇聚项，可消除密集 $O(N)$ 掩码分配和键值复制开销，同时保持精确的局部注意力。

- **采用 256 项查找表的 GQA 打包 2 位 Metal 内核**`half4`**查找表：** 通过一次共享内存查表对每个通道的 4 个坐标进行反量化，并将所有 $r = 3$

查询头保存在寄存器中，采用 Block-8 惰性 exp2 重缩放，达到 73.4GB/s，性能优于 bf16（3.36× 在 256K 时）和 4 位 q4_0（2.60× 在 1M 时）。

- **实证验证：** 我们验证了 $F_{\text{关闭}} \leq 0.41$位（表现良好，未发生旋转偏移崩溃），在 2.06 位时的 $+0.130\,\Delta$困惑度（在 3.06 位时为 -0.007，在 2.56 位时为 $+0.024$），以及在 8K、16K 和 32K 下 100% 精确检索 7 位通行密钥。

II. 架构分析与偏移安全性

Inkling-2-Mini 使用 $n_q = 24$个查询头、 $n_{\text{kv}} = 8$个键值头（GQA 比率为 $r = 3$）、头维度为 $d = 128$，并进行 48 次逻辑层访问（6 个前奏层、 $+ 3 \times 12$ 个循环核心层和 + 6个尾奏层）。每个 6 层模块包含 5 个滑动窗口层（$W = 1024$）和 1 个全局层（总计 8 次全局访问）。不同于 RoPE Transformer，位置信息通过逐通道短卷积（宽度为 4）和范围为 $R = 1024$ 的加性相对偏置 $r_t W_{\text{rel}}$ 进行编码，此外还使用了 s_t依赖查询的汇聚项对数几率。

A. 偏移脆弱性探测（F_{off}）

对于键在缓慢旋转的 RoPE 维度上带有较大恒定偏移的模型，逐 token KV 量化会彻底失效（例如 Qwen2.5 k-bias 或 Qwen3-small 的 QK-norm 增益放大 [5]）。在量化 Inkling-2-Mini 之前，我们在 2,048 个 WikiText-2 词元的全部 48 次逻辑访问中，测量了一次遍历的偏移浪费统计量 $F_{\text{off}} = \max_{l,\,h} \log_2(1 + 0.674\,\rho_{l,\,h}/3.5)$，并使用 $\rho = \|\mu_k\|/\text{rms}(k - \mu_k)$。每次访问的得分均为 $F_{\text{off}} \in [0.18, 0.41]$ 位（远低于 1.3 位的良性阈值），因为 Inkling-2 没有 RoPE、没有 K 偏置，采用 L2 裁剪的 Q/K，且 V 的 RMS 为单位值。因此，旋转后的逐 token KV 量化无需进行偏移中心化。

B. 原生运行时为何无法应对长上下文

审查 mlx-vlm 和 oMLX 后，我们发现了两项叠加性低效问题：

1)**逐 token KV 复制以实现学习型汇聚点（mlx-vlm）：** 每个解码 token 都通过 mx.concatenate 追加一个零 K/V 向量，会复制整个 KV 缓存。向 scaled_dot_product_attention 传入 MLX 原生的 sinks= 参数，在位级上完全一致（0.00e+00 maxdiff），并且在 1K 到 128K 范围内快 1.85–2.44× faster。

2)**无界滑动缓存（oMLX）：** 为全部 48 次访问分配 KVCache()，会存储 48 段完整历史，而不是 8 段。将其中 40 次滑动访问替换为 RotatingKVCache（为 DSpark 推测验证预留 +16 token 的回滚余量），可将 32K KV 内存占用从 6.05 GiB 降至 1.36 GiB（4.45×），并修复推测式环绕回滚问题（已合并/提交为 oMLX PR #5512）。

III. 无尺度 2 位 SRFT 压缩与 Metal 内核

A. 正交归一 *SRFT* 与无尺度单位均方根值

对于任意头向量 $x \in \mathbb{R}^d$，我们应用实数正交归一化子采样随机傅里叶变换：

$$\tilde{x} = \text{SRFT}(x) = \mathcal{F}_{\mathbb{R}}(S \odot x), \qquad S \in \{-1, +1\}^d, \qquad (1)$$

其中，$\mathcal{F}_{\mathbb{R}}$ 将归一化的实数快速傅里叶变换输出 $[X_0/\sqrt{d}, X_{d/2}/\sqrt{d}, \sqrt{2}/d\,\overline{\text{Re}}(X_{1..d/2-1})\ \sqrt{2}/d\,\text{Im}(\overline{X_{1..d/2-1}})$打包在一起。由于 SRFT 严格正交（$\langle \text{SRFT}(q), \text{SRFT}(k) \rangle = \langle q, k \rangle$ 至 1.14×10^{-5}），解码注意力直接在旋转域中运行：每个查询仅变换q 一次，并在每次全局访问时仅对单个输出向量执行一次逆变换 o_{far} 一次，无需对缓存键或值执行任何逆快速傅里叶变换。

经过 SRFT 后，坐标近似服从高斯分布且同分布，因此标量单位高斯 Lloyd–Max 质心 C_b 近乎最优（在 $b = 2$: $C_2 = \{-1.5104, -0.4528, +0.4528, +1.5104\}$ 时，均方误差为 $= 0.1175\sigma^2$）。关键在于，Inkling-2 对 V进行无参数 RMS 归一化（使用_值_范数_无_gamma=True），从而保证 rms($\tilde{v}$) =rms(v) $= 1$ 恒定。因此，V无需存储**任何尺度参数**（精确为每坐标 b 位），而 K 则为每个 128 元素向量存储一个 fp16 RMS 范数（每坐标 b+0.125 位），由此得到：

- SRFT lm2：2.125（K） + 2.000（V）有效位数（7.76× 对比 bf16）。
- SRFT K3/V2：3.125（K）+2.000（V） $\Rightarrow$ **2.5625** 有效位数（6.24×）。
- SRFT lm3：3.125（K） + 3.000（V） $\Rightarrow$ **3.0625** 有效位数（5.23×）。

与 OSCAR 不同， [6],这种方法无需离线校准；与 TurboQuant 不同， [4], 它避免将 1 位用于一个 QJL 符号残差（在位数预算相同的情况下，这会增大 softmax logit 方差）。

B. 分割带状偏置注意力与精确在线对数和指数合并

对于序列长度为 $N > 1024$ 的每次全局访问，加性相对偏置仅 在最新的 $R = 1024$个键上非零。因此，我们将全局缓存划分为：

1) **远端前缀** $[0, N - 1024)$**：** 通过 SRFT 分成 256 个令牌块溢出，并打包为 2 位编码（ K 每行 32 字节， V 每行 32 字节，另加 K 尺度的每头每令牌 = 66 字节）。通过我们的 GQA 打包 Metal 内核解码，返回（o_{far}， ℓ_{far}），无需掩码内存。

2) **近期窗口** $[N - 1024, N)$ **+学习型汇聚项：** 以精确的 bf16 格式保留，使用精确的 1,024 宽带状相对偏置和依赖查询的汇聚项对数几率进行计算，得到 ($o_{\text{near}}, \ell_{\text{near}}$)，并通过精确的双路对数和指数合并与 ($o_{\text{far}}, \ell_{\text{far}}$) 合并：

$$m = \max(\ell_{\text{far}}, \ell_{\text{near}}), \quad o = \frac{e^{\ell_{\text{far}} - m}\, o_{\text{far}} + e^{\ell_{\text{near}} - m}\, o_{\text{near}}}{e^{\ell_{\text{far}} - m} + e^{\ell_{\text{near}} - m}}. \qquad (2)$$

C. GQA 打包 *2* 位 *Metal* 内核 *(kv_lm2_metal)*

对于 $d = 128$ 和每坐标 2 位的情况，每个 32 车道 Metal SIMD 组会将 4 个坐标映射到每个车道：**每个车道 1 字节**，从而为每个 token 产生完全合并的 32 字节突发传输。我们的内核消除了通用 Metal 注意力的三项开销：

1) **零 GQA 填充浪费：** 此前的 simdgroup_matrix 内核会将 $r = 3$ 填充至 8 行（浪费 62.5% 的计算）。

我们的内核会将所有 $r = 3$ 个查询头（half4 q0、q1、q2）保存在寄存器中，每个 K 和 V 字节只反量化一次，并同时更新全部 3 个头。

2) **256 项线程组 half4 查找表：**每个字节编码 4 个二位 Lloyd–Max 索引（共 256 种状态）。在内核入口处预先计算 2 KiB 线程组 half4 查找表[256]，即可通过一次查表解码全部 4 个坐标，取代 28 条移位/掩码/选择指令，并将寄存器压力减半。

3) **按块处理 8 个令牌的延迟 exp2 缩放：**每个头先计算 8 个令牌分数，再执行一次硬件 exp2 缩放，从而降低缩放开销 $8\times$。

在每个键值头上划分为 64–256 个分区进行 Split-K 并行，使 M6 的 GPU 核心达到饱和，随后执行一次精确的 LSE 归约调度。与基于反量化代码的 float32 参考注意力（$N = 4096$）相比，该内核的最大输出误差为1.39×10^{-4}，最大 LSE 误差为 1.46×10^{-4}。

IV. 在 32GB Apple M6 上的评估

所有测量均在 32 GB Apple M6 Mac mini（DRAM 带宽 150GB/s，Metal 工作集 26.8 GiB）上使用 mlx 0.32.4 和 mlx-community/Inkling-2-Mini-oQ4e-mtp（15.10 GiB 活动权重）完成。

A. 从 *4K* 到 *1,048,576* 词元的端到端速度与内存

表 I 和图 1 报告了从 4K 到 1M 词元的端到端解码吞吐量、KV 缓存占用和 GPU 峰值内存。在 256K 时，m6_lm2_srft 的吞吐量达到 **17.09 令牌/秒**（比 m6_bf16 的 5.09 令牌/秒快 $3.36\times$），同时将 KV 内存减少$6.5\times$（1.27 GiB，而非 8.20 GiB）。在完整的 **1,048,576 词元**原生上下文下，m6_lm2_srft 的解码速度为**9.52 令牌/秒**（105.1 毫秒/词元），总峰值内存为**19.49 GiB**（KV 为 4.37 GiB）——比 4 位 q4_0 SRFT（3.66 令牌/秒，24.36 GiB）快 $2.60\times$，且 KV 占用比原版 oMLX 小 $44\times \sim$193GiB。

B. 内核优化阶梯与 *SRFT* 和 *Hadamard* 的比较

图 2 展示了在各轮内核迭代中，生成 100 万个令牌时的解码速度提升：从 4.5 位 q4_0 SRFT（3.66 令牌/秒），到采用逐令牌 exp() 的朴素 2 位 LM2（6.67 令牌/秒，41.7 GB/s），再到加入 Block-8 惰性 exp2 重缩放（8.37 令牌/秒，59.4 GB/s），以及引入包含 256 个条目的 half4 查找表（9.52 令牌/秒，73.4 GB/s；生成 100 万个令牌时每层耗时 7.65 毫秒）。有趣的是，在 64K–256K 时，m6_q4_srft 的速度也超过了内核内 FWHT（64K 时为 20.85 对 18.88 令牌/秒；128K 时为 17.52 对 15.65 令牌/秒），这是因为使用 MLX 的 Metal FFT 仅对单个查询/输出向量进行旋转，使注意力内核无需内核内蝶形屏障即可运行，同时保持相同的质量（两者在 4K 时的贪心令牌一致率均为 63/64）。

C. 语言建模质量与 *8K–32K* 通行密钥检索

表 II 评估多窗口 WikiText-2 困惑度（$4\times$4096 个词元；对每个窗口最后 2,048 个词元进行评分，较早的前缀则存放在量化远端缓存中）

表 I：在 32 GB Apple M6 上端到端运行 Inkling-2-Mini 解码

上下文	Mode	令牌/秒	毫秒/令牌	键值（GiB）	峰值（GiB）
4K	原版 oMLX	25.27	39.58	0.80	15.91
4K	m6_q4_srft (4.5b)	25.31	39.50	0.47	15.59
32K	原版 oMLX	19.36	51.66	6.05	21.16
32K	m6_bf16 (PR #5512)	19.71	50.74	1.36	16.47
32K	m6_q4_srft (4.5b)	22.24	44.96	0.72	15.84
64K	m6_bf16	16.55	60.42	2.20	17.32
64K	m6_q4_srft (4.5b)	20.85	47.96	0.81	15.93
128K	m6_bf16	11.59	86.30	4.20	19.33
128K	m6_q4_srft (4.5b)	17.52	57.09	1.38	16.50
256K	m6_bf16	5.09	196.40	8.20	23.34
256K	m6_q4_srft (4.5b)	12.50	80.01	2.50	17.62
256K	m6_lm2_srft **(2.06b)**	**17.09**	**58.51**	**1.27**	**16.39**
512K	m6_q4_srft (4.5b)	8.42	118.82	4.75	19.87
512K	m6_lm2_srft **(2.06b)**	**13.81**	**72.40**	**2.31**	**17.43**
1M	m6_q4_srft (4.5b)	3.66	272.94	9.20	24.36
1M	m6_lm2_srft **(2.06b)**	**9.52**	**105.08**	**4.37**	**19.49**

表 II 质量（4×4096 WIKITEXT-2）和长程 NIAH 检索

格式	有效位数	PPL	ΔPPL	8K	16K	32K
bf16 (参考)	16.000	6.7489	+0.0000	通过	通过	通过
SRFT lm4	4.0625	6.7428	−0.0061	—	—	—
SRFT lm3	3.0625	6.7418	−0.0071	通过	PASS	PASS
SRFT K3/V2	2.5625	6.7725	+0.0236	通过	通过	通过
SRFT lm2	**2.0625**	**6.8785**	**+0.1296**	**通过**	**通过**	**通过**

同时评估长距离大海捞针（NIAH）检索：一个 7 位通行密钥被埋在 8K、16K 和 32K WikiText-2 干草堆中位置 1,500 处——距查询 6.7K 至 30.5K 个词元，对 40 个滑动层和 1,024 个词元的 bf16 局部窗口而言完全不可见。

SRFT lm3（3.06 位）与 bf16（Δ困惑度 $= -0.0071$）几乎无法区分；SRFT K3/V2（2.56 位）仅增加$+0.0236$（$+0.35\%$），而 SRFT lm2（2.06 位）增加 $+0.1296$（$+1.9\%$），同时在最长 32K 的上下文中通过了**3/3 项精确的 7 位数字通行密钥检索**。

D. 关于稀疏混合专家上的推测解码

在 M6 上测试 Inkling-2-Mini 随附的 DSpark 起草器（$B = 8$）后，我们发现了混合专家的专家并集权衡：验证 8 个词元会读取 128 个专家中的 $\sim$35–45 个不同专家的并集，耗时 120 毫秒，而验证 1 个词元仅需 40 毫秒。固定的 $k = 7$ 可提升代码性能（$24.80 \rightarrow 27.52$ 令牌/秒，每步 3.76 个词元，128/128 完全匹配），但会降低散文性能（$25.78 \rightarrow 22.90$令牌/秒，每步 2.93 个词元）。将起草结果量化为 4 位，并通过接受率指数移动平均自适应调整验证深度 $k \in \{2, 3, 5, 7\}$，可消除散文性能下降（26.15 令牌/秒），并将代码性能提升至 28.61 令牌/秒（128/128 完全匹配）。

V. 相关工作与局限性

逐通道键值量化（KIVI [7],KVQuant [8]）和基于旋转的方法（QuaRot [9],TurboQuant [4], OS- CAR [6]）面向全注意力 RoPE 模型，并且要么需要离线校准（OSCAR），要么需要承担 1 位 QJL 开销

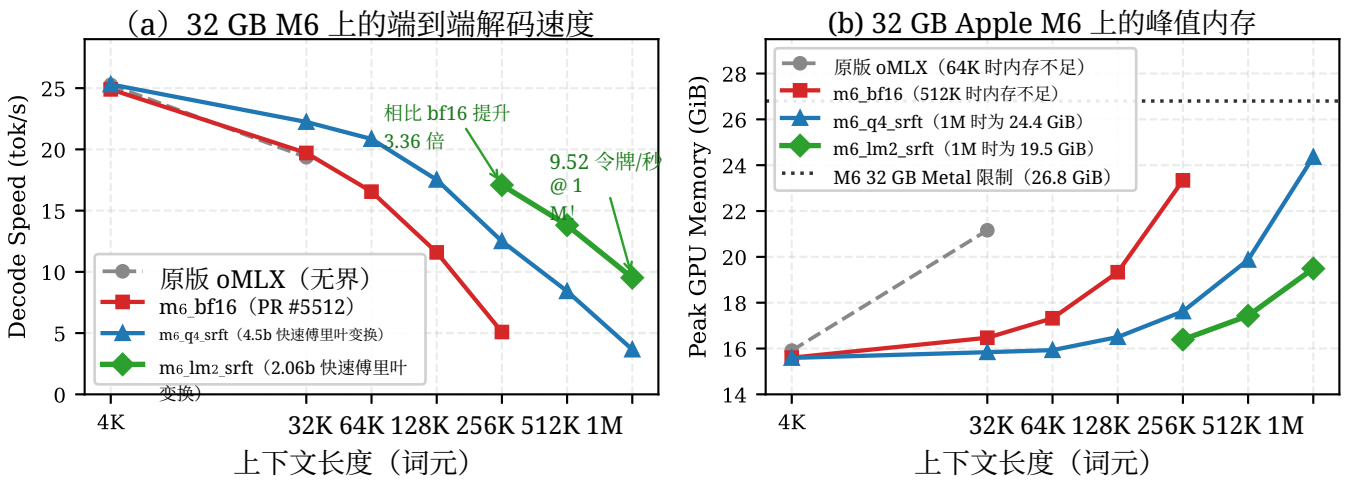

图 1。在 32 GB Apple M6 上，从 4K 到 1,048,576 个词元的 Inkling-2-Mini（26.4B 循环式混合专家模型）端到端解码速度和峰值 GPU 内存。原版 oMLX 在约 64K 处因内存不足而退出；滑动固定 `m6_bf16` 在 512K 之前因内存不足而退出。打包的 2.06 位 SRFT（`m6_lm2_srft`）在 256K 时达到 17.09 令牌/秒（3.36× 相比 bf16），并以 9.52 令牌/秒在 19.49GiB 内存中完成完整的 1M 上下文。

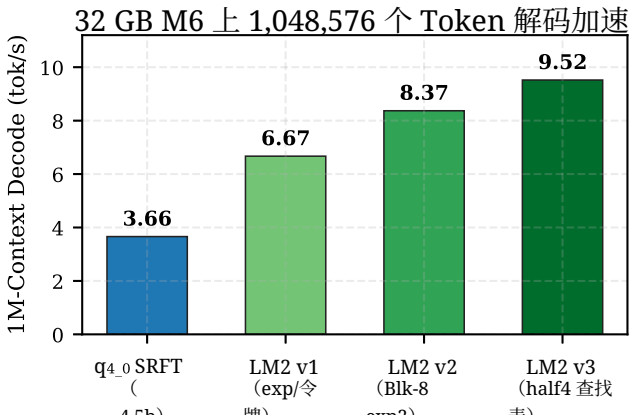

图 2。在 32GB M6 上，跨越内核优化的 1,048,576 个词元端到端解码进程（相较于 4 位 SRFT 的 2.60× 总加速比）。

（TurboQuant）。我们的研究表明，采用单位 RMS 值的现代 NoPE 混合架构适用于完全无需校准、无需尺度参数的 2 位 SRFT 压缩，同时加速统一内存上的解码。

局限性。端到端 256K–1M 解码计时采用直接填充缓存，而非耗时数小时的预填充；NIAH 的评估长度上限为 32K（每种长度仅使用一个针）；质量评估基于 WikiText-2 和通行密钥检索，而非完整的 RULER 或长链推理基准测试。

VI. 结论

通过结合有界滑动窗口缓存、分段带偏置的在线 LSE 合并、无需尺度参数的 2.000 位 SRFT 值压缩，以及 half4-LUT GQA 打包 Metal 内核，我们得以在 32GB Apple M6 Mac mini 上，以 9.52 令牌/秒对 26.4B 循环式混合专家模型进行完整**1,048,576 令牌推理，耗用 19.49GiB** ——3.36× 在 256K 时比 bf16 更快，且 44× 更小键值占用比原版 oMLX 更少——在最高 32K 的范围内，通行密钥检索率达 100%，且无需离线校准。

可复现性。所有 Metal 内核、基准测试和交互式聊天脚本均已在 MIT 许可证下开源，地址为 https://github.com/aminems/inkling-1m-mlx [10]，；滑动窗口缓存修复已向上游提交，地址为 https: // github.com/jundot/omlx/pull/5512。

参考文献

[1] Thinking Machines Lab，《Inkling 2 Mini：开放权重循环式多模态混合专家模型》，https://huggingface.co/thinkingmachines/ Inkling-2-Mini，2026 年 11 月。[2] oMLX 贡献者，《oMLX：适用于 Apple Silicon 的优化 MLX 推理引擎》，https://github.com/jundot/omlx，2026 年。[3] M. A. Bergach，《量化无需成本：在 Apple Silicon 上超越 fp16 的 int4 键值缓存》，2026 年。[4] A. Zandieh、M. Daliri、M. Hadian 和 V. Mirrokni，《TurboQuant：具有近最优失真率的在线向量量化》，载于 国际学习表征会议（*ICLR*），2026 年。[5] A. Jonasson，《大型语言模型中的旋转偏移特征》，2025 年。[6] Z. Zhou、D. Zhuang、J. Li、Z. Chen、S. L. Song、B. Athiwaratkun 和 X. Wu，《OSCAR：用于 2 比特键值缓存量化的离线谱协方差感知旋转》，2026 年。[7] Z. Liu、J. Yuan、H. Jin、S. Zhong、Z. Xu、V. Braverman、B. Chen 和 X. Hu，《KIVI：无需调优的键值缓存非对称 2 比特量化》，载于 国际机器学习会议（*ICML*），2024 年。[8] C. Hooper、S. Kim、H. Mohammadzadeh、M. W. Mahoney、Y. S. Shao、K. Keutzer 和 A. Gholami，《KVQuant：面向千万级上下文长度大型语言模型推理的键值缓存量化》，载于 神经信息处理系统进展（*NeurIPS*），2024 年。[9] S. Ashkboos、A. Mohtashami、M. L. Croci、B. Li、P. Cameron、M. Jaggi、D. Alistarh、T. Hoefler 和 J. Hensman，《QuaRot：旋转后的大语言模型中的无异常值 4 比特推理》，载于 神经信息处理系统进展（*NeurIPS*），2024 年。[10] M. A. Bergach，《inkling-1m-mlx：面向 Apple Silicon 上 1M 上下文 Inkling-2-Mini 的无需校准 2 比特 SRFT 键值缓存压缩》，https:// github.com/aminems/inkling-1m-mlx，2026 年。