SunBURST：以众数为中心的拉普拉斯积分，实现确定性的图形处理器加速贝叶斯证据计算

Ira Wolfson[1]

[1]以色列布劳德工程学院电子与电气工程系

2026年1月29日

arXiv:2601.19957v1 [stat.CO] 27 Jan 2026

摘要

由于维数灾难以及基于采样的方法具有顺序执行的特性，在高维空间中评估贝叶斯证据的计算成本高得难以承受。我们提出 SunBURST，这是一种用于贝叶斯证据计算的确定性 GPU 原生算法，以众数为中心的几何积分取代了全局体积探索。该流程结合了径向众数发现、批量 L-BFGS 精细化和基于拉普拉斯方法的解析积分，对各个众数分别处理，并将大批量似然评估转化为高度并行的 GPU 工作负载。

对于高斯后验和近高斯后验，在这些情形下拉普拉斯近似是精确的或高度准确的；在我们的基准测试中，SunBURST 在最高达 1024 维时仍能达到双精度容差下的数值一致性，并且在测试范围内，墙钟时间呈次线性扩展。在多峰高斯混合分布中，保守配置可实现低于百分之一的误差，同时保持良好的扩展性。

SunBURST 并非旨在普遍取代基于采样的推断。它的设计面向物理参数估计和反问题中常见的情形：后验质量在有限个众数附近可由高斯结构局部近似。在强非高斯情形下，该方法可用作快速的几何感知证据估计器，或作为混合工作流的预处理阶段。这些结果表明，通过将确定性积分与大规模并行相结合，可以使超高维中的高精度贝叶斯证据评估在计算上切实可行。

关键词：贝叶斯推断、证据计算、嵌套采样、GPU 计算、高维积分

1 引言

贝叶斯模型比较需要计算边际似然或证据 e:

$$\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)\,\mathrm{d}\theta \tag{1}$$

其中， $\mathcal{L}(\theta) = p(D|\theta)$ 是似然， $\pi(\theta)$ 是先验。竞争模型之间的证据之比即为贝叶斯因子，为模型选择提供了一种基于原则的框架，并通过奥卡姆剃刀自动惩罚复杂性 [1, 2]。在宇宙学中，基于证据的模型比较已成为区分暴胀情景、暗能量参数化以及对 ΛCDM [3] 的扩展的重要手段。

计算 $\mathcal{Z}$ 一直以难度极高著称。该积分遍及整个先验体积，而先验体积会随维度 d 呈指数增长；与此同时，后验质量却集中在这个体积中呈指数级缩小的一小部分区域内——这就是众所周知的维数灾难 [4]。传统蒙特卡洛方法难以应对，因为随机采样几乎不可能命中相关区域；重要性采样要求提议分布预先近似后验分布；而热力学积分则需要在从先验到后验的整个过渡过程中仔细安排温度。

1.1 嵌套采样

嵌套采样 [5, 6]巧妙地将多维证据积分转换为关于先验质量 X的一维积分:

$$\mathcal{Z} = \int_0^1 \mathcal{L}(X)\,\mathrm{d}X \tag{2}$$

其中 $X(\lambda) = \int_{\mathcal{L}(\theta)>\lambda} \pi(\theta)\,\mathrm{d}\theta$ 表示由等似然轮廓 $\mathcal{L} = \lambda$所包围的先验体积。该变换将问题从对复杂后验分布进行采样，转化为在跟踪似然阈值的同时迭代压缩先验体积。

现代实现已在天体物理推断中取得显著成功。Multi-Nest [7, 8] 针对多模态后验分布引入了椭球分解。PolyChord [9]以切片采样取代拒绝采样，使效率提升至 $d \sim 50$。较新的实现包括采用动态活动点分配的 dynesty [10] 、强调稳健性的 UltraNest [11]，以及利用 JAX 实现 GPU 加速的 JAXNS [12] 。

尽管取得了这些进展，嵌套采样仍面临根本性的扩展性局限。计算成本按 $O(n_{\mathrm{live}} \cdot n_{\mathrm{iter}})$ 扩展，其中 $n_{\mathrm{iter}} \propto n_{\mathrm{live}} \cdot H$ 表示后验信息 H，而对于 d 维度中的 K 个众数，其成本为 $n_{\mathrm{live}} = O(K \cdot d)$。这导致 $O(K \cdot d^2)$ 的扩展性，在维度超过 $d \approx 50$–100 后变得难以承受。

1.2 图形处理器带来的机遇

图形处理器可提供强大的并行能力，但嵌套采样中活动点的顺序收缩，阻碍了直接加速。每次迭代都依赖于上一次的似然阈值，因此图形处理器的加速比仅限于并行执行单次似然评估。

对于能够用高斯混合分布良好近似的后验分布，我们发现，证据积分可以采用一种截然不同的计算策略:

1. 通过并行优化检测后验众数 2. 精化众数位置并估计局部曲率 3.
通过拉普拉斯近似计算各众数的证据 [13]4. 汇总各众数的贡献

每一步都可轻松并行，从而充分利用图形处理器。

1.3 贡献

我们推出了 SunBURST（**S**eeded **U**niverse **N**avigation—**B**ayesian **U**nification via **R**adial**S**hooting **T**echniques），这是一款图形处理器加速的证据计算器，可实现：

1. $O(K)$ **模态扩展**：证据按众数计算，仅需 $O(K)$ 个众数表征，而非 $O(K \cdot d)$ 个活动点
2. **次线性墙钟时间扩展性**：总体 $O(d^{0.7})$扩展性从 2维扩展至 1024维，在消费级 GPU 上处理 1024维数据时（针对高斯基准测试），绝对耗时低于 7 秒 3. **自动模态检测**：批量优化无需预先了解众数的数量或位置即可发现后验众数 4. **数值精度**：对于高斯后验，在 1024维以内相对误差低于 10^{-12}

该流程包含三个核心模块，其名称取自广平杨式太极拳招式，以此向Donald Rubbo大师致敬：

- **抱虎归山** （抱虎归山，$Bào\ Hˇu\ Gu¯ı\ Sh¯an$—“抱虎归山”）：通过多尺度光线投射和批量 L-BFGS 优化，并结合收敛—反收敛振荡进行模态检测

- **GreenDragon** （青龙出水，$Q¯ıng\ Lóng\ Ch¯u\ Shuˇi$—“青龙出水”）：使用图形处理器并行 L-BFGS 进行峰值精化 [14] ，结合海森矩阵估计与鞍点筛选

- **BendTheBow** （弯弓射虎，$W¯an\ G¯ong\ Shè\ Hˇu$—“弯弓射虎”）：通过拉普拉斯近似计算证据，并针对一般高斯几何结构采用自适应海森矩阵估计

SunBURST 并非旨在普遍取代基于采样的证据估计器。其设计面向这样一种情形：后验质量在有限数量的众数附近可由高斯结构进行局部良好近似——这种情形常见于物理参数估计、模型校准和逆问题。在强非高斯情形下（例如明显的曲率、相变或重尾结构），应将 SunBURST 视为一种快速的几何感知证据估计器，或作为混合推断工作流中的预处理阶段。

为支持此类全面的方法，可选的第四个模块 **GraspBirdsTail(揽雀尾)** 会执行基于特征分解的降维，以便在需要完整后验表征时移交给下游采样器。

SunBURST 使用 CuPy [15] 在 Python 中实现 GPU 加速，并可自动回退到 NumPy [16] 和 SciPy [17] ，以适用于仅配备中央处理器的系统。

2 维数灾难

在介绍我们的算法之前，我们必须面对一个根本障碍：它使高维贝叶斯证据计算变得极其困难。“维数灾难”这一术语由 Bellman [18] 首创，指的是随着维度数量增加而出现的一系列病态现象。在证据计算中，这些病态现象相互作用，使问题的难度呈指数级增长，其影响不仅仅是带来不便，而是真正具有毁灭性。

熟悉这些病态现象的读者可以跳到第 2.7 节，查看 SunBURST 必须克服的主要挑战概要。

2.1 测度集中

考虑一个 d 维超立方体 $[-1,1]^d$，其体积为 $V_d = 2^d$。其内切半径为 1 的超球体体积为：

$$V_{\text{sphere}}(d) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \tag{3}$$

其中 $\Gamma(z)$ 是伽马函数，即阶乘的连续延拓：对于正整数，有 $\Gamma(n) = (n-1)!$，且 $\Gamma(1/2) = \sqrt{\pi}$。导致维数灾难的关键性质是 $\Gamma(d/2+1)$ 超指数增长——其增长速度快于 $\pi^{d/2}$，从而导致球体体积趋近于零。

比值 $V_{\text{sphere}}/V_{\text{cube}}$ 揭示了第一个令人震惊的现象：

d	2	10	50	100	500	1000
$V_{\text{sphere}}/V_{\text{cube}}$	0.79	2.5×10^{-3}	1.9×10^{-24}	1.9×10^{-70}	$\sim 10^{-616}$	$\sim 10^{-1464}$

到了 100 维时，球体在立方体中所占的比例，比质子体积与可观测宇宙体积之比还小。到了 500 维时，我们已经耗尽了双精度浮点数的动态范围（10^{-308}）。高维空间的几何结构由角落主导。

2.2 壳层定理

高维高斯分布的概率质量分布在哪里？对于 $\mathcal{N}(0,\ I_d)$，概率质量的径向分布为：

$$p(r) \propto r^{d-1} e^{-r^2/2} \tag{4}$$

该分布在 $r^* = \sqrt{d - \overline{1 \approx}}\sqrt{d}$ 处达到峰值，标准差为 $\sigma_r \approx 1/\sqrt{2}$，且与 d 无关。概率质量集中在一个薄壳中：

d	壳层半径 r^*	壳层厚度 σ_r	相对厚度
10	3.0	0.71	24%
100	10.0	0.71	7.1%
1000	31.6	0.71	2.2%

高维高斯分布的众数（位于原点）包含可忽略的概率质量。几乎所有概率都集中在半径为 $\sqrt{d}$、厚度为 $O(1)$ 的壳层中——相对于包围球而言，这是一个测度为零的集合，因为 $d \to \infty$。

2.3 距离灾难

两点之间的欧几里得距离 $x, y \in \mathbb{R}^d$ 为：

$$d(x,y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{5}$$

考虑单位超立方体 $[0,1]^d$中均匀分布的两个随机点。每一项（$x_i - y_i$）^{2} 都是一个均值为 $\mu = \mathbb{E}[(x_i - y_i)^2] = 1/6$ 且方差有限的独立随机变量。这类项的和共有 d项，均值为 $d/6$，因此：

$$\mathbb{E}[d(x,y)] \approx \sqrt{d/6} \xrightarrow{d\to\infty} \infty \tag{6}$$

任意两 个随机点之间的期望距离会发散。但总和的标准差仅以 $\sqrt{d}$ 的速度增长，因此变异系数（标准差除以均值）趋于零：

$$\frac{\mathrm{Std}[d(x,y)]}{\mathbb{E}[d(x,y)]} \sim \frac{\sqrt{d}}{d} = \frac{1}{\sqrt{d}} \to 0 \tag{7}$$

到最近邻和最远邻的距离都会集中在 $\sqrt{d/6}$ 附近。极限情况下，所有点到其他所有点的距离都相等。“近”和“远”的概念失去了意义。聚类算法会失效。最近邻搜索变得随意。我们依赖的 2维和 3维直觉中的几何根本不存在。

The L_∞ 度量。 L_∞ 度量不会遭受 L_2 的发散病态 ：

$$d_\infty(x,y) = \max_{i=1,\dots,d} |x_i - y_i| \tag{8}$$

由于这是最大值而非总和，因此它有界：对于 $[0,1]^d$ 中的点，无论维度如何，我们都有 $d_\infty(x, y) \in [0,1]$。然而，近期研究表明，随着 $d \to \infty$，随机点之间的 L_∞ 距离仍会集中到一个共同值，使最近邻搜索变得不可靠 [19]。对于 SunBURST 的应用场景——去重已收敛到相似位置的峰值，以及相对于已知参考点检测收敛——L_∞ 仍然有效，因为我们比较的是来自优化轨迹的结构化点，而非均匀随机样本。

2.4 采样灾难

要以网格间距 δ 均匀覆盖一个 d 维单位超立方体，我们需要 $N = (1/\delta)^d$ 个点。对于 $\delta = 0.1$（每个维度 10 个点）：

d	网格点	10^9/秒	存储（64 位）
10	10^{10}	10 秒	80 GB
20	10^{20}	3,000 年	8×10^{11} TB
50	10^{50}	3×10^{33} 年	—
100	10^{100}	3×10^{83} 年	—

在 100 个维度上以 0.1 为间距建立网格，所需点数超过可观测宇宙中的原子数量（$\sim 10^{80}$）。蒙特卡洛方法也好不到哪里去：要通过朴素抽样将积分的相对误差控制在 ϵ，需要 $N = O(\epsilon^{-2})$ 个样本才能实际命中相关区域。当后验分布仅占先验体积的 $f \sim 10^{-70}$ 时，我们需要 $N \sim 10^{70}$ 个样本，才会有一个落入后验分布中。

2.5 证据计算灾难

对于贝叶斯证据，这些病态问题会叠加。证据积分：

$$\mathcal{Z} = \int_\Theta \mathcal{L}(\theta)\pi(\theta)\, d\theta \tag{9}$$

需要在整个先验体积 V_Θ 上进行积分，但后验质量集中在一个指数级狭小的区域内。压缩比——先验体积与后验体积之比——呈指数级增长：

$$\frac{V_{\text{prior}}}{V_{\text{posterior}}} \sim \left(\frac{\sigma_{\text{prior}}}{\sigma_{\text{posterior}}}\right)^d \tag{10}$$

对于每个维度上都比后验分布宽的先验 $10\times$，该比值为 10^d。在 $d = 100$ 时，我们必须对占据先验体积 10^{-100} 的区域进行精确积分。

嵌套采样通过先验质量变换解决了这一问题，但付出了不同的代价：迭代次数按 $n_{\text{iter}} \sim n_{\text{live}} \cdot H$ 进行缩放，其中信息量 $H = \int p(\theta|D) \log[p(\theta|D)/\pi(\theta)]\, d\theta$ 通常按 $O(d)$ 增长。再加上 K 个众数对应的 $n_{\text{live}} = O(K \cdot d)$，总成本变为 $O(K \cdot d^2)$——仍然是多项式级别，但增长十分陡峭。

2.6 多峰灾难

多个后验众数会将困难转变为不可能。如果 K 个众数被低似然谷分隔开，嵌套采样就必须维持足够多的活动点，以填充每个众数。所需的 $n_{\text{live}} = O(K \cdot d)$ 可确保每个众数至少有一个活动点，但在 d 较高时，这将变得难以承受。

更糟的是，模态检测本身也变得不可靠。在高维空间中，随机采样几乎永远无法发现众数——落在任意给定点的 ϵ 距离范围内的概率按 ϵ^d 缩放。若不了解众数位置，算法就只能采用成本高昂的探索阶段。

2.7 总结：维数灾难的五个方面

维数灾难通过五种截然不同但彼此相关的病态现象表现出来，任何高维证据计算器都必须克服这些问题：

1. **体积坍缩**：超球体与超立方体的体积比以超指数速度趋于零 ly—，当维度为 $d = 100$ 时，球体仅占立方体的 $< 10^{-70}$，浮点精度会在维度达到 $d \approx 500$ 时耗尽。2. **壳层集中**：概率质量从众数迁移到壳层。一个 d维高斯分布集中在半径为 $\sqrt{d}$ 、厚度为 O 的壳层中 (1)；众数所含的概率质量可以忽略不计。3. **距离集中**：在 L_2 下，所有两两距离都收敛到 $\sqrt{d}/6$—— 最近邻与最远邻变得无法区分。 L_∞ 可避免发散，但随机点的距离仍会集中。4. **采样爆炸**：网格覆盖需要 $(1/\delta)^d$ 个点；蒙特卡洛需要 $O(f^{-1})$ 个样本，其中 $f \sim 10^{-d}$ 是后验体积与先验体积之比。5. **积分灾难**：证据计算必须对占据先验体积 $\sim 10^{-d}$ 的后验分布进行积分。嵌套采样的成本为 $O(K \cdot d^2)$；朴素方法则需要指数级计算量。

这些病态问题分别引出了相应的设计约束，第 3 节所述的 SunBURST 算法对此进行了明确处理。

2.8 逃逸几何

任何算法如何才能突破这些指数障碍？维数灾难不仅仅是计算上的不便，它反映了真实的数学结构。没有任何算法能够高效地对高维空间进行均匀采样。

逃逸路径如下：

1. **利用结构**：如果后验分布具有特殊结构（例如近似高斯、可分离、稀疏），就充分利用这些结构。2. **局部处理**：放弃全局覆盖，将计算资源集中于后验质量所在区域。3. **并行化**：通过图形处理器以并行广度换取顺序深度。4. **重用样本**：绝不丢弃似然评估，而是将其保存起来，供后续阶段使用。SunBURST 运用了这四种策略。下文将介绍具体做法。

3 算法

3.1 概述

SunBURST 通过预处理步骤和随后执行的三个主要模块来计算贝叶斯证据：

- **预处理步骤**：通过 $O(2d)$ 敏感性探测，检测并移除病态维度（平坦、柔性的干扰因素）

- **模块 1（CarryTiger）**：通过在先验超立方体中进行射线投射来发现后验众数，随后通过伴随收敛—反收敛振荡的 ChiSao 探索进行搜索
- **模块 2（GreenDragon）**：通过批量 L-BFGS 将峰值位置细化至误差在 $< 10^{-12}$ 以内，并生成 TrajectoryBank
- **模块 3（BendTheBow）**：通过拉普拉斯近似计算各众数的证据，并利用 TrajectoryBank 实现零成本旋转检测

第四个可选模块（$\mathbf{GraspBirdsTail}$）通过特征分解进行降维，以便交由下游采样器处理。

样本库会在 GPU 内存中保存所有似然评估结果，从而无需重复计算即可进行几何检测。

3.1.1 预处理

在进行模态检测之前，我们会探测似然，以检测病态维度：

- **平坦维度**：敏感性为零——似然完全忽略这些参数
- **软干扰维度**：约束极弱，会导致条件数问题

对病态维度进行解析边缘化，剩余的“活跃”维度则进入模块 1。

3.1.2 模块 1：CarryTiger

我们使用四种互补的几何方式在先验超立方体中投射射线（顶点到顶点、顶点到边、面到面，以及从中心发出的 SunBURST[1] ）。沿每条射线，我们分两个阶段进行采样：

1. **粗采样阶段**：通过均匀采样，找出似然不可忽略的区域
2. **细化阶段**：在粗采样阶段发现似然显著的区域集中进行自适应重采样

从有希望的射线样本出发，ChiSao 探索通过收敛—反收敛振荡定位后验众数。到达真实峰值的样本会“粘住”并保持冻结，而其他样本则继续探索。

[1] 一种以某个内部点为中心进行射线投射的方法，类似于从太阳向外发射射线，因此得名。

3.1.3 模块2：GreenDragon

模块 1 得到的粗略峰值位置通过以下方式细化，使误差控制在 10^{-12} 以内：

1. 在每个粗略峰值周围设置多个起始点 2. 使用严格收敛容
差的批量 L-BFGS 3. 去重和鞍点筛选

精确的峰值位置至关重要：拉普拉斯近似误差会随其与真实最大值之间的距离增大而增大。L-BFGS 轨迹会存储在 TrajectoryBank 中，供模块 3 使用。

3.1.4 模块3：BendTheBow

对于每个精化后的峰值，我们通过拉普拉斯近似计算其证据贡献：

1. **几何检测**：检查后验分布是否存在非对角相关性（使用 Green- Dragon 的 TrajectoryBank 或有限差分探测） 2. **海森矩阵计算**：对于轴对齐情况仅计算对角元素，对于旋转后的后验分布使用完整矩阵 3. **拉普拉斯近似**：使用海森行列式进行解析高斯积分 总证据是所有众数上的证据之和（对数和指数）。

3.2 问题陈述

给定对数似然函数 $\ell(\theta) = \log \mathcal{L}(\theta)$，以及定义在 d 维超立方体 $\Theta = \prod_{i=1}^{d}[a_i, b_i]$ 上的均匀先验 $\pi(\theta)$，我们要计算对数证据：

$$\log Z = \log \int_{\Theta} \exp(\ell(\theta))\, d\theta - \log V_{\Theta} \tag{11}$$

先验假设。 当前实现假设似然完整地包含在先验超立方体内，即先验边界不会截去任何显著的似然质量。在此假设下，先验体积 $V_{\Theta} = \prod_i (b_i - a_i)$ 仅作为归一化常数。如何处理与似然存在非平凡交互的信息性先验或非均匀先验，将留待未来研究。

3.3 模块结构

模块名称		输出
-	预处理	活跃维度，边缘修正
1	CarryTiger	粗略峰值 + RayBank
2	GreenDragon	精化峰值 + 海森矩阵 + TrajectoryBank
3	BendTheBow	对数 Z
4	GraspBirdsTail	约化似然（可选）

样本库将所有似然评估持久保存在 GPU 内存中：

- **RayBank**：射线几何、样本位置、来自模块 1（CarryTiger）的对数似然
- **TrajectoryBank**：来自模块 2（GreenDragon）的 L-BFGS 轨迹，用于模块 3 中的旋转检测

这些组能够在模块 3 中进行几何检测，而无需额外计算似然。

3.4 预处理：维度病态检测

在模态检测之前，SunBURST 会进行快速的 $O(2d)$ 探测，以检测可能导致后续阶段出现数值困难的病态维度。

3.4.1 敏感性探测

对于每个维度 i，我们通过从先验中心扰动至边界，评估似然敏感性：

$$s_i = |\ell(\theta_{\text{center}} + \Delta_i e_i) - \ell(\theta_{\text{center}})| + |\ell(\theta_{\text{center}} - \Delta_i e_i) - \ell(\theta_{\text{center}})| \tag{12}$$

其中，$\Delta_i = (b_i - a_i)/2$ 是维度 i 上先验的半宽，e_i 是第 i 个单位向量。这恰好需要 $2d$ 次似然评估。

3.4.2 平坦维度

似然完全不敏感的维度：

$$s_i < \epsilon_{\text{flat}} \cdot \text{median}(s_1, \ldots, s_d) \tag{13}$$

默认值为 $\epsilon_{\text{flat}} = 10^{-6}$。这些参数是数据完全无法约束的参数——后验分布等于先验。

3.4.3 软干扰维度

约束非常弱、会导致条件数问题的维度：

$$s_i < \epsilon_{\text{soft}} \cdot \max(s_1, \ldots, s_d) \quad \text{and} \quad s_i \geq \epsilon_{\text{flat}} \cdot \text{median}(s) \tag{14}$$

默认值为 $\epsilon_{\text{soft}} = 10^{-3}$。此阈值会捕获条件数 $\kappa > 30$。

3.4.4 解析边缘化

被标记的维度会从活动问题中移除，并进行解析边缘化：

平坦维度对证据的贡献为 $\log(\Delta_i)$（均匀先验积分）。

软干扰维度贡献一个高斯积分。我们根据敏感性估算有效宽度σ_i：

$$\sigma_i \approx \frac{\Delta_i}{\sqrt{s_i}}, \quad \text{contribution} = \frac{1}{2}\log(2\pi) + \log(\sigma_i) \tag{15}$$

干扰维度边缘化所依据的假设。应用于“软干扰”维度的高斯边缘修正，假设沿每个此类坐标，对数似然在探测参考点（先验中心）附近可由一维二次函数很好地近似。在此假设下，中心与先验边界之间测得的对数似然下降可作为局部曲率的替代指标，从而估算有效高斯宽度 σ_i。由此得到的贡献 $\frac{1}{2}\log(2\pi) + \log \sigma_i$ 对应于对该坐标上的局部高斯因子进行解析边缘化。

当似然在此尺度上并非近似二次函数，或先验中心在该坐标上远离局部最大值时，应将此修正理解为一种

启发式尺度估计，而非精确的边缘化。在这种情况下，SunBURST 的主要拉普拉斯阶段（使用检测到的众数处的曲率）仍是证据精度的主要来源。

剩余的 d_{active} 维度将交由 CarryTiger 处理。此预处理可防止后续海森矩阵计算中的条件数灾难性恶化。

3.4.5 算法摘要

算法 1 预处理：维度病态检测 要求：对数似然 ℓ、边界 $[a_i, b_i]$、阈值 $\epsilon_{\text{flat}}, \epsilon_{\text{soft}}$确保：活跃维度、边缘修正 1: $\theta_{\text{center}} \leftarrow (a+b)/22$: $\ell_0 \leftarrow \ell(\theta_{\text{center}})$ 3: 对于 $i = 1, \ldots, d$ 执行▷图形处理器并行：在一个批次中计算所有 $2d$ 4: $\Delta_i \leftarrow (b_i - a_i)/25$: $s_i \leftarrow |\ell(\theta_{\text{center}} + \Delta_i e_i) - \ell_0| + |\ell(\theta_{\text{center}} - \Delta_i e_i) - \ell_0|$6: 结束循环 7: 平坦 $\leftarrow \{i : s_i < \epsilon_{\text{flat}} \cdot$中位数$(s)\}$8: 软 $\leftarrow \{i : s_i < \epsilon_{\text{soft}} \cdot$最大值$(s)\} \setminus$平坦 9: 活跃 $\leftarrow \{1, \ldots, d\} \setminus$(平坦 $\cup$软) 10: 对数 $Z_{\text{marginal}} \leftarrow \sum_{i \in}$平坦 对数$(\Delta_i) + \sum_{i \in}$软 $\left[\frac{1}{2}\right.$对数$(2\pi) +$对数$(\Delta_i/\sqrt{s_i})\left.\right]$11: 返回活跃、对数 Z_{marginal}

3.5 模块 1：CarryTiger（抱抱抱虎虎虎归归归山山山– 抱虎归山）

CarryTiger 通过射线投射和 ChiSao 探索发现后验众数 .

3.5.1 射线生成

四种射线类型为先验超立方体提供互补覆盖（图 1） :

- **顶点到顶点 (V2V)**：连接随机顶点对——这些射线贯穿完整的对角线范围
- **顶点到边 (V2E)**：从顶点朝向边的中点——中间角度
- **面到面（W2W）**：在相对面上的随机点之间——保证沿某一坐标方向覆盖整个范围
- **Sunburst**：从超立方体中心沿均匀分布的方向发出 S^{d-1}——实现各向同性覆盖

射线数量随维度调整： $n_{\text{rays}} = 10 + \log_2(d)$。

3.5.2 两阶段采样

沿每条射线 $r(t) = p_{\text{start}} + t \cdot (p_{\text{end}} - p_{\text{start}})$，针对 $t \in [0, 1]$:

粗采样阶段。 在 n_{coarse} 个点处均匀采样。计算对数似然。确定满足阈值 Δ 的“活跃区域”，其中 $\ell(\theta) > \ell_{\max} - \Delta$ 。

细化阶段。 按活动区域内与似然成正比的密度重新采样。这会将评估集中在关键区域。

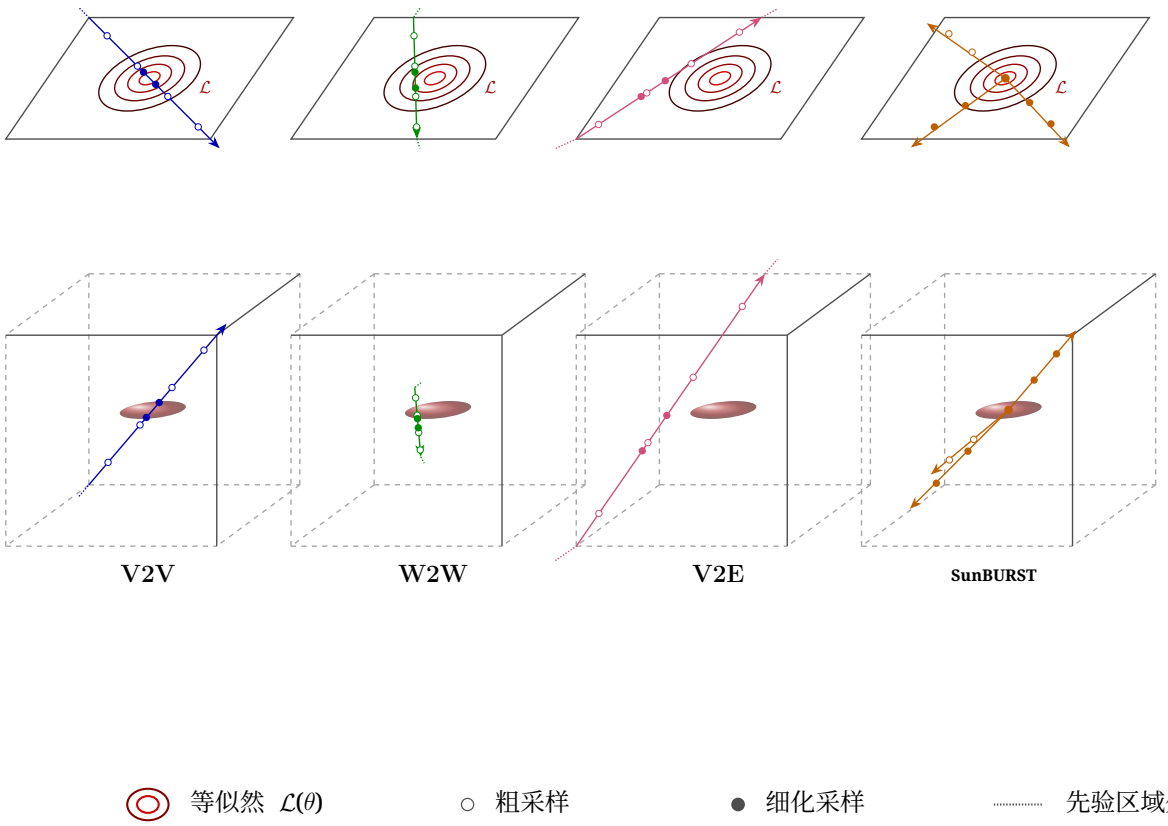

图 1：用于发现似然质量的射线投射策略。上排：显示等似然线的二维投影。下排：包含似然区域（红色椭球体）的三维先验超立方体。**V2V**：顶点到顶点射线贯穿完整对角线。**W2W**：面到面射线连接相对面。**V2E**：顶点到边射线提供中间角度。**Sunburst**：射线从中心向四面八方辐射。空心圆表示粗略的均匀采样；实心圆表示集中在似然区域附近的细化采样。

3.5.3 通过 SingleWhip 发现尺度（單單單鞭鞭鞭）

探索前，我们使用曲率分析，根据射线样本估计特征尺度。对于沿射线的对数似然样本 $\ell(t)$，我们在多个跳过长度 h 处计算二阶导数：

$$\kappa(h) = \max_t \left| \frac{\ell(t+h) - 2\ell(t) + \ell(t-h)}{h^2} \right| \tag{16}$$

当 $\kappa(h)$ 急剧下降时，表示存在特征尺度 $\lambda_{\text{fine}}, \lambda_{\text{mid}}, \lambda_{\text{coarse}}$。

3.5.4 ChiSao（黟黟黟手手手 –黟手）探索

ChiSao 实现了振荡式收敛—反收敛策略。这个名称指的是咏春的“黟手”技巧——找到真实峰值的样本会“黏住”并保持冻结，而其他样本则继续探索。

振荡周期。对于 `n_`振荡个周期（默认 3）：

1. **收敛阶段**：释放所有停滞掩码，并运行批量 L-BFGS，朝局部极大值迭代 n_{converge} 次。

2. **停滞检测**：满足两个 条件的样本会被标记为“停滞”：

   - 梯度 L_∞ 范数低于容差（已收敛）
   - 对数似然$\ell \geq \ell_{\text{最大值}}+\log(0.1)$，即似然至少达到当前最大值的 10%

   似然阈值可防止停滞在质量较差的局部极大值处。

3. **去重**：使用 L_∞ 度量对峰值进行去重，移除 K_{lost} 重复样本。保留似然最高的样本；其逆海森矩阵估计将用于宽度估计。

4. **倒攆猴 (**倒倒倒攆攆攆猴猴猴**) / 金鸡独立 (**金金金雞雞雞獨獨獨立立立**)**：如果 $K_{\text{lost}} > 0$ 且不是最后一次振荡，则维持样本数量：

   - **标准情况 (**≥ 5未收敛样本**)**：倒攆猴从未收敛样本出发，沿随机方向发射射线以重新播种
   - **接近耗尽 (**< 5 个未收敛样本**)**：金鸡独立利用正交射线方向（通过 QR 分解）从已收敛的 峰值补充样本，发挥图形处理器的并行能力

   新样本会被标记为未受阻，以便进行探索。如果新样本未发现任何新峰值，金鸡独立会自适应地停用。

5. **云手 (**雲雲雲手手手**)**：如果不是最后一次振荡，未受阻样本会执行 n_{cloud}步梯度上升，使用随机平滑的 梯度：

$$\nabla \ell_\sigma(\theta) \approx \frac{1}{K} \sum_{k=1}^{K} \nabla \ell(\theta + \sigma \cdot z_k), \quad z_k \sim \mathcal{N}(0, I_d) \tag{17}$$

   这会模糊小于 σ 的特征，通过局部噪声揭示全局结构。对于多尺度问题（Rastrigin 函数、Ackley 函数），这使样本能够透过局部波纹观察到全局盆地。如果未提供平滑尺度 σ ，则会根据样本分布范围自动估算。

6. **反收敛阶段**：脱困的样本会接受基于$n_{\text{anticonverge}}$ 动量的梯度下降 步：

$$v_{t+1} = \mu \cdot v_t - \alpha \cdot \nabla \ell(\theta_t), \quad \theta_{t+1} = \theta_t + v_{t+1} \tag{18}$$

使用动量 $\mu = 0.9$ 和步长 α。动量使样本能够“越过”谷底并发现新的峰值。在此阶段，停滞的样本会始终冻结在其峰值处。

设计依据。 该顺序是经过刻意安排的：

- **收敛后去重**：样本在收敛后聚集程度最高，此时进行重复项检测最为有效
- **重新播种前进行平滑**：新样本能够从云手阶段中受益
- **反收敛前进行平滑**：在基于动量的探索开始前，将样本推向全局吸引域
- **最后一次振荡时跳过步骤 4–6**：最终收敛会生成输出峰值，无需进一步扰动

3.5.5 算法摘要

算法 2 CarryTiger：通过 ChiSao 探索进行众数发现 要求：射线样本 $\{x_i\}$、对数似然 $\{\ell_i\}$、参数 n_{osc}、n_{conv}、n_{anti}确保：粗略峰值、RayBank 1：从顶部射线样本初始化样本 x 2：停滞 $\leftarrow$假^N 3：对于 $k = 1$、.....、n_{osc} 执行 4：停滞 $\leftarrow$假^N ▷ 释放全部 5：x、$\nabla \ell \leftarrow$ L-BFGS(x、n_{conv}) ▷ 收敛 6：停滞 $\leftarrow (\|\nabla \ell\|_\infty < \epsilon) \wedge (\ell \geq \ell_{\max} +$ 对数 0.1) ▷ 停滞检测 7：$x \leftarrow$ 去重(x、L_∞) ▷ 移除重复项 8：如果 $k < n_{\text{osc}}$ 则 9：$x \leftarrow$ 重新播种(x、停滞) ▷ 倒撵猴 / 金鸡独立 10：$x[\neg$停滞$] \leftarrow$ HLC($x[\neg$停滞$]$) ▷ 云手 11：$x[\neg$停滞$] \leftarrow$ 反收敛($x[\neg$停滞$]$、n_{anti}) ▷ 动量下降 12：结束如果 13：结束循环 14：返回 x[停滞],RayBank

3.5.6 输出

模块 1 返回 K 粗略峰值位置及宽度估计，另附包含所有射线样本和对数似然的 RayBank。

3.6 模块 2：GreenDragon（青龙出水– **青龙出水**）

GreenDragon 将粗略峰值位置精细化至机器精度。

3.6.1 种子初始化策略

对于每个粗略峰值（位于 θ^*，估计宽度为 w）：

- **默认模式**：每个轴在 $\theta^* \pm 0.9w \cdot e_i$ 处设置 $2d$ 个样本作为种子 i
- **快速模式**：设置 20 个随机扰动作为种子

轴对齐种子支持 GPU 并行对角海森矩阵计算。所有 $2d$ 扰动均通过一次批量 GPU 调用进行评估：

$$H_{ii} = \frac{\ell(\theta^* + he_i) - 2\ell(\theta^*) + \ell(\theta^* - he_i)}{h^2} \tag{19}$$

这样即可在 $O(1)$ 的实际运行时间成本下得到完整的对角海森矩阵（假设有 $P \gg d$ 个 GPU 核心），而顺序评估需要 $O(d)$。

3.6.2 批量 L-BFGS 优化 t

所有峰值上的所有初始种子均同时进行优化。迭代次数会自适应：$\max(10, 3\log_2(d))$。

3.6.3 轨迹存储

L-BFGS 轨迹存储在 TrajectoryBank 中。每条轨迹都会记录优化过程中访问的位置序列，使模块 3（BendTheBow）无需额外进行似然评估，即可分析优化路径以检测旋转。这是实现零成本几何检测的关键机制。

3.6.4 筛选：鞍点与非驻点

两个滤波器确保只有真正的极大值点得以保留：

鞍点滤波器。 任何具有 $H_{ii} > 0$（正曲率表示上升方向）的候选点都是鞍点，会被移除。

梯度滤波器。 具有 $\|\nabla \ell\|_\infty > \epsilon_{\text{grad}}$ 的候选点并非真正的驻点，因此会被移除。L-BFGS 可“免费”提供梯度范数，无需额外进行似然评估。

3.6.5 算法摘要

算法 3 GreenDragon：峰值精化

要求： 模块 1 提供的粗略峰值 $\{\theta_k^*\}$ 及其宽度 $\{w_k\}$
确保： 精化峰值、对角海森矩阵、TrajectoryBank

对于每个峰值 k **执行** ▷ 通过图形处理器在所有峰值间并行处理
  在 $\theta_k^* \pm 0.9w_k \cdot e_i$ 处为 $i = 1$ 初始化 $2d$ 个样本, . . . , d
结束循环
$x, \nabla\ell$,通过 L-BFGS 优化轨迹 ← （所有种子, 返回_轨迹 = True）
将轨迹存入 TrajectoryBank
根据种子评估计算对角海森矩阵 H_{ii} ▷已计算
是_最大值 ← 所有($H_{ii} < 0$, 轴 $= 1$) ▷鞍点筛选
是_驻点 ← $\|\nabla\ell\|_\infty < \epsilon_{\text{grad}}$ ▷ 梯度滤波器（自由）
保留 ← 是_最大值 ∧ 是_驻点
对保留的峰值去重
返回精化峰值, $\{H_{ii}\}$, TrajectoryBank

3.6.6 输出

模块 2 返回 K' 个精化峰值及其对角海森矩阵 $\{H_{ii}\}$，并返回包含所有 L-BFGS优化路径的 TrajectoryBank。TrajectoryBank 是模块 3（BendTheBow）进行旋转检测的主要输入。

3.7 模块 3：BendTheBow（弯弯弯弓弓弓射射射虎虎虎 – 弯弓射虎 r)

BendTheBow 通过拉普拉斯近似计算贝叶斯证据。

3.7.1 拉普拉斯近似

对于每个经细化的峰值 θ_k^* 及其海森矩阵 H_k:

$$\log Z_k = \ell(\theta_k^*) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\det(-H_k)| \tag{20}$$

对于高斯后验而言，这是精确的；对于近高斯后验而言，则能提供出色的近似。

3.7.2 自动几何检测

一个关键挑战是检测后验分布是否存在需要使用完整海森矩阵的非对角相关性。模块 2 仅提供对角线 $\{H_{ii}\}$ 在 O（d）的计算成本；完整海森矩阵的计算成本为 O（d^2）。SunBURST 采用两步检测:

步骤 1：垂直步长比例（无需额外评估次数）。分析 GreenDragon's TrajectoryBank 中的 L-BFGS 轨迹。在白化坐标中（按 $\sqrt{-H_{ii}}$ 缩放），计算每条轨迹中每一步的:

$$f_\perp = \frac{\|\Delta\theta_\perp\|}{\|\Delta\theta\|} \tag{21}$$

其中， $\Delta\theta_\perp$ 是垂直于梯度方向的分量。对于轴对齐后验分布，优化沿坐标轴进行（$f_\perp \approx 0$）。明显的垂直运动（$f_\perp > \epsilon_{\text{rot}}$）表明存在旋转。

步骤 2：有限差分探测（2 次评估）。如果基于轨迹的检测无法得出结论，则直接探测非对角结构:

$$b = \frac{v^T H v - a}{d-1}, \quad v = \frac{1}{\sqrt{d}}[1,1,\ldots,1]^T \tag{22}$$

其中， a =均值（H_{ii}）。显著的 $|b|$ 表明存在均匀相关性。

Full 海森矩阵计算。如果检测到旋转，则通过中心差分计算:

$$H_{ij} = \frac{\ell_{++} - \ell_{+-} - \ell_{-+} + \ell_{--}}{4\epsilon^2} \tag{23}$$

这需要进行 $O(d^2)$ 次似然评估。在配备 $P \gg d^2$ 个核心的图形处理器上，所有 $2d^2$ 个扰动都通过一次批处理调用进行评估，实际运行时间为 $O(d)$。

典型与最坏情况复杂度。完整海森矩阵的计算在最坏情况下的成本为 $O(d^2)$，而 SunBURST 在典型情况下会检测轴对齐结构，仅计算对角曲率，从而避免这一成本；只有检测到旋转时才会计算完整海森矩阵。这种条件式方法带来：

$$\text{Hessian cost} = \begin{cases} O(d) & \text{axis-aligned posteriors (typical)} \\ O(d^2) & \text{rotated posteriors (rare)} \end{cases} \tag{24}$$

3.7.3 证据计算

对角海森矩阵（轴对齐后验分布）。

$$\log Z_k = \ell(\theta_k^*) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{d}\log(-H_{ii}) \tag{25}$$

完整海森矩阵（一般后验分布）。通过 GPU 加速特征分解计算：

$$\log|\det(-H)| = \sum_{i=1}^{d}\log(\lambda_i) \tag{26}$$

其中 $\{\lambda_i\}$ 是 $-H$的特征值。

3.7.4 多模态证据

总证据是所有 K 个众数的证据之和：

$$\log Z = \text{logsumexp}(\log Z_1, \ldots, \log Z_K) \tag{27}$$

3.7.5 算法摘要

算法 4 BendTheBow证据计算 要求：峰值 $\{\theta_k^*\}$、来自 GreenDragon 的对角海森矩阵 $\{H_k\}$、来自 GreenDragon 的 TrajectoryBank 确保：对数证据 log Z1：对于每个峰值 $k = 1, \ldots,$ K 执行 2：旋转 ←垂直_步长_比例（轨迹k） 3：如果无法确定，则 4：旋转 ← 探测_非对角项（θ_k^*, H_k）▷ 2 次评估 5：结束如果 6：如果已旋转，则 7：H_k ←完整_海森矩阵（θ_k^*） 8：log $Z_k \leftarrow \ell_k + \frac{d}{2}$log(2$\pi$) $- \frac{1}{2}$log |det($-H_k$)|9：否则 10：log$Z_k \leftarrow \ell_k + \frac{d}{2}$log(2$\pi$) $- \frac{1}{2}\sum_i$log($-H_{ii}$) 11：结束如果 12：结束循环 13：返回 logsumexp(log $Z_1, \ldots,$ log Z_K)

▷ $O(d^2)$ evals

– –

– –

3.8 模块 4：GraspBirdsTail (揽雀尾 – 揽雀尾) [可选]

GraspBirdsTail 为下游采样器（Poly-Chord、dynesty、emcee）准备压缩似然，以便交接使用。此模块为可选模块；当需要进行完整的后验采样时，它会在证据计算后运行。

3.8.1 特征分解分析

使用主峰处的完整海森矩阵 H （或根据轨迹库构建的近似矩阵），计算特征分解：

$$-H = V\Lambda V^T, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d) \tag{28}$$

特征值揭示了后验几何结构，并对各个方向进行分类：

- **信息方向**：$\lambda_i > \epsilon_{\text{有信息}}$ — 受到数据的强约束
- **干扰方向**：$\epsilon_{\text{干扰}} < \lambda_i < \epsilon_{\text{有信息}}$ — 受到的约束较弱
- **退化方向**：$\lambda_i < \epsilon_{\text{干扰}}$ — 不受约束（平坦）

3.8.2 降维

有效维数 d_{eff} 计数信息方向：

$$d_{\text{eff}} = \#\{i : \lambda_i > \epsilon_{\text{inform}}\} \tag{29}$$

旋转矩阵 V 变换到主坐标，其中：

- 信息方向成为前 d_{eff} 个坐标
- 干扰方向可以通过解析方式边缘化（高斯积分）
- 退化方向的贡献为其先验宽度

3.8.3 压缩似然

约化似然 $\ell_{\text{red}}(\phi)$ 仅作用于 d_{eff} 个维度：

$$\ell_{\text{red}}(\phi) = \ell(V_{\text{info}}^T \phi + \theta^*_{\text{nuisance}}) \tag{30}$$

其中， V_{info} 包含信息特征向量， $\theta^*_{\text{nuisance}}$ 将干扰参数固定为其最大后验值。

3.8.4 算法摘要

算法 5 GraspBirdsTail：采样器交接的降维**需要：** 主峰 θ^*、海森矩阵 H、阈值 ϵ_{inform}，$\epsilon_{\text{nuisance}}$**确保：** 约化似然、旋转矩阵、边缘贡献1: V， $\Lambda \leftarrow \mathrm{eig}(-H)$ ▷ 特征分解

n

2: 信息 $\leftarrow \{i : \lambda_i > \epsilon_{\text{inform}}\}$3: 干扰 $\leftarrow \{i : \epsilon_{\text{nuisance}} < \lambda_i \leq \epsilon_{\text{inform}}\}$4: 退化 $\leftarrow \{i : \lambda_i \leq \epsilon_{\text{nuisance}}\}$5: $d_{\text{eff}} \leftarrow$ |信息|6: $V_{\text{info}} \leftarrow V[:, \text{信息}]$7: $\ell_{\text{red}}(\phi) \leftarrow \ell(V_{\text{info}}^T \phi + \theta^*_{\text{nuisance}})$8: 对数 $Z_{\text{nuisance}} \leftarrow \sum_{i \in \text{干扰}} \frac{1}{2} \log(2\pi/\lambda_i)$9: 对数 $Z_{\text{degen}} \leftarrow \sum_{i \in \text{退化}} \log(\Delta_i)$10: **返回** ℓ_{red}、 V、 d_{eff}、 $\log Z_{\text{nuisance}} + \log Z_{\text{degen}}$

3.8.5 输出

GraspBirdsTail 导出：

- 约化似然 $\ell_{\text{红色}}(\phi)$ 在 $d_{\text{有效}}$ 维度上
- 用于坐标变换的旋转矩阵 V
- 约化空间中的边界
- 干扰/退化方向的边缘贡献

即使原始问题包含许多干扰方向，这也能实现高效的后验采样，因为下游采样器是在约化的 d_{eff} 维空间中运行的。

3.9 扩展性策略

维数灾难无法消除——这是高维空间的一项基本属性。不过，SunBURST 采用了多种策略来延缓其出现并减轻其影响：

3.9.1 峰值附近的径向积分

我们围绕检测到的众数进行积分，而不是对整个先验体积积分。指数级庞大的先验体积仅作为解析归一化常数出现。

3.9.2 GPU 并行化

区分计算复杂度 （似然评估次数）与墙钟时间扩展性 （在并行硬件上的耗时）非常重要：

操作	评估次数	CPU 墙钟时间	GPU 墙钟时间
梯度（有限差分）	$2d$	$O(d)$	$O(1)$
对角海森矩阵	$2d$	$O(d)$	$O(1)$
完整海森矩阵	$2d^2$	$O(d^2)$	$O(d)$
L-BFGS 迭代	$O(d)$	$O(d)$	$O(1)$

在 GPU并行性充足的情况下，完整海森矩阵所需的 $2d^2$ 次扰动可以通过少量大型批处理调用完成计算。算法 复杂度仍为 $O(d^2)$，但在图形处理器吞吐量饱和的范围内，实际运行 时间可更接近 $O(d)$ 的扩展性，并且内核启动开销得以摊销。

3.9.3 $O(K)$ 模态扩展

嵌套采样在理论上的扩展性为 $O(K \cdot d)$ 个活动点，但由于受约束先验采样的难度，其实际性能会随维度增加而急剧下降 [9, 20]。SunBURST 通过局部拉普拉斯近似独立处理各个众数，无论维度如何，都只需 $O(K)$ 个有效样本。

3.9.4 样本复用

每次似然评估都会存入数据库。模块 3 重用 GreenDragon's TrajectoryBank 进行几何检测，在常见情况下实现零成本旋转诊断。

3.10 计算复杂度

方法	海森矩阵计算成本	备注
拉普拉斯（对角）	$O(d)$	仅轴对齐
拉普拉斯（完整）	$O(d^2)$	一般后验
黎曼流形哈密顿蒙特卡罗	每步 $O(d^2)$	每步更新度量
嵌套采样	$O(d^2)$ 隐式	活动点提议
SunBURST（典型情况）	$O(d)$	旋转检测无需完整海森矩阵
SunBURST（最坏情况）	$O(d^2)$	旋转后的后验分布

对于包含 K 个众数的多模态问题：

方法	评估次数	GPU 墙钟时间
嵌套采样	$O(K \cdot d^2 \cdot H)$	$O(K \cdot d^2 \cdot H)$
SunBURST（典型）	$O(K \cdot d)$	$O(K \cdot d^{0.5})$
SunBURST（旋转后）	$O(K \cdot d^2)$	$O(K \cdot d)$

其中，H 是信息增益（通常为 $\propto d$）。实测的 $O(d^{0.5})$ 墙钟时间扩展性反映了 $O(d)$ 算法复杂度的 GPU 并行化。

请注意，嵌套采样中经常引用的扩展性表达式可能低估高维情况下的端到端成本。在实际实现中，每次迭代都需要在不断变化的似然等高线内进行受约束先验采样，并执行几何更新（例如协方差估计和边界构造），其复杂度至少为 d的二次方。随着维数增加，提出有效受约束样本的难度通常会成为主导因素，并可能导致实际运行时间的扩展性显著劣于二次方。Skilling [6] 给出了理想化的 $O(d^2)$ 成本项，而 Handley 等人 [9] 报告了 PolyChord 在现实条件下的 $O(d^3)$ 扩展性；Ashton 等人 [20] 的综述讨论了不同实现中渐近描述与实际性能之间长期存在的差距。

准确性问题。嵌套采样得出的证据估计对活动点数量、提议/边界策略和终止准则十分敏感；在有限预算下，报告的内部不确定性未必经过良好校准。即使在高斯测试问题上，已发表的结果也报告了不可忽视的证据不确定性（例如，在典型的 PolyChord 设置下，以对数 Z 单位计为 ~ 0.4）[9]。Buchner [21] 开发了诊断测试，显示嵌套采样证据估计存在系统性偏差；Higson 等人 [22] 记录了特定于实现的失效模式，其中后验探索不完整，导致证据估计出现偏差；Nelson 等人 [23]则报告了 MultiNest 中与维度相关的偏差。JAXNS 基准测试 [12]还显示，在标准配置下，MultiNest 在中等维度的高斯似然问题上会失效。

相比之下，SunBURST 直接针对高斯分布和近高斯后验分布进行校准证据估计。对于严格的高斯后验分布（包括偏心、雪茄形、相关以及旋转雪茄形几何结构），拉普拉斯近似是精确的；在我们的基准测试中，SunBURST 在 512维–1024维范围内达到浮点容差级别的数值一致性。对于高斯混合，准确度取决于配置：采用保守设置（3 次振荡和完整的 GreenDragon 精化）时，4 众数混合函数在 512维下的误差为 0.07%；而快速配置以准确度换取速度（例如在 768维下误差为 1.4%）。

更广泛地说，后验采样与证据校准对算法有不同的要求；在后验探索中的准确度，并不意味着在实际计算预算下能够在高维度中自动实现校准的 $\log Z$ 。

3.11 局限性

拉普拉斯近似假设后验分布在每个众数附近都可以用高斯分布很好地近似。对于高斯后验，这种近似是精确的；在许多物理学应用中也很准确。

对于具有重尾、强偏度或香蕉形退化的后验分布，拉普拉斯近似可能会引入偏差。

当前局限性包括：

- **先验包含假设**：似然必须完全包含在先验超立方体内；目前尚未处理先验边界造成的截断
- **仅支持均匀先验**：非均匀先验或信息性先验需要未来扩展支持
- **非高斯后验**：拉普拉斯方法无法捕捉重尾和偏度
- **香蕉形退化**：弯曲脊线不符合局部高斯假设
- **相变**：似然中的尖锐特征可能会被射线投射遗漏

通过重要性采样校正、考虑先验的积分或混合方法来扩展 SunBURST，以处理此类情况，仍是未来的研究方向（第 7.1 节）。

4 理论基础

本节为 SunBURST 的三项关键创新奠定理论基础：径向射线模态发现、证据计算的拉普拉斯近似，以及图形处理器并行化的复杂度。

4.1 通过径向射线进行模态检测

SunBURST 的第一阶段通过从先验边界向内部发射射线来发现后验众数。我们将形式化描述该方法在何种条件下能够保证定位到众数。

对数凹性。 概率密度 $p(\theta)$ 若 $\log p(\theta)$ 是凹函数，则称其为对数凹的，即对于所有 θ_1、θ_2 和 $\lambda \in [0,1]$:

$$\log p(\lambda\theta_1 + (1-\lambda)\theta_2) \geq \lambda \log p(\theta_1) + (1-\lambda)\log p(\theta_2) \tag{31}$$

许多常见的后验分布都是对数凹的，包括高斯分布、指数分布、逻辑斯蒂分布和拉普拉斯分布，以及由具有对数凹先验 [24] 的广义线性模型产生的后验分布。

径向射线模态检测。 设 $p(\theta)$ 是定义在凸域 $\Omega \subset \mathbb{R}^d$ 上、在 θ^* 处具有唯一模态的单峰对数凹密度。对于边界点 $p \in \partial\Omega$ 和指向 Ωd 内部的单位方向 $\hat{}$，射线 $r(t) = p + td$ 满足：(i) 对数似然剖面 $\ell(t) = \log p(r(t))$ 关于 t 是单峰的；(ii) 最大值出现在使 $\ell(t)$ 达到最大值的取值 t^* 处，其中 $r(t^*)$ 位于连接 p 与 θ^* 的线段上。

这是因为凹函数在直线上的限制仍然是凹函数。由于 $\log p$ 在 Ω 上是凹函数，因此函数 $\ell(t) = \log p(r(t))$ 在满足 $r(t) \in \Omega$ 的区间内关于 t 是凹函数。凸域上的凹函数至多有一个局部最大值，而该最大值必定是全局最大值。

对于多模态后验分布，对数凹性假设在全局范围内不成立，但通常在每个众数附近局部成立。这促使 CarryTiger 算法（第 3.5 节）进行局部最大值检测：每条射线可能会出现多个局部最大值，分别对应沿射线经过的不同众数；通过寻找沿射线方向的似然梯度发生符号变化的点，即可识别这些众数。

这些保证不适用于支集非连通、狭窄脊线与射线方向正交，或似然结构高度振荡的后验分布；ChiSao 的随机重新播种在经验上能够处理这类情况，但不提供形式完备性保证。

非凸后验分布。当后验分布并非全局对数凹时，径向射线可能会出现多个峰值。SunBURST 通过 ChiSao 振荡（第 3.5.4 节）来处理这一情况：在多轮扫描中随机改变射线方向，以确保覆盖所有重要众数。

4.2 证据的拉普拉斯近似

给定模态位置 θ^*，SunBURST 使用拉普拉斯近似计算证据贡献。

拉普拉斯近似。 设 $p(\theta|D) \propto \exp(-nf(\theta))$，其中 f 是光滑函数，并在 θ^* 处有唯一极小值，且海森矩阵 $H = \nabla^2 f(\theta^*)$ 为正定。则 [13]:

$$\int_{\mathbb{R}^d} \exp(-nf(\theta))\, d\theta = \exp(-nf(\theta^*)) \cdot \frac{(2\pi)^{d/2}}{n^{d/2}|H|^{1/2}} \cdot \left(1 + O(n^{-1})\right) \tag{32}$$

取对数后，得到 SunBURST 中使用的证据近似：

$$\log \mathcal{Z} \approx \log \mathcal{L}(\theta^*) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{H}| \tag{33}$$

其中 $\mathbf{H} = -\nabla^2 \log \mathcal{L}(\theta^*)$ 是观测信息矩阵（在众数处对数似然的负海森矩阵）。

高斯后验的精确性。 当 $\log\ \mathcal{L}(\theta) = -\frac{1}{2}(\theta-\mu)^\top\Sigma^{-1}(\theta-\mu)+c$ 时，拉普拉斯近似是精确的：泰勒展开中的所有高阶项都恒为零。这解释了 SunBURST 在高斯测试函数上达到机器精度准确度的原因。

误差界。 对于非高斯后验， $O(n^{-1})$ 误差项取决于 f 在众数 [2] 处的三阶和四阶导数。偏度和峰度较小（形状接近高斯分布）的后验分布误差较小。

多模态扩展。 对于具有 K 个相互分离的众数、位于 $\{\theta_k^*\}_{k=1}^K$ 的后验分布，总证据分解为：

$$\mathcal{Z} = \sum_{k=1}^{K} \mathcal{Z}_k \approx \sum_{k=1}^{K} \exp\left(\log \mathcal{L}(\theta_k^*) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{H}_k|\right) \tag{34}$$

当各众数相距足够远，使其高斯近似之间的重叠可以忽略不计时，这种拉普拉斯混合近似是准确的；对于定义良好的多峰后验，这一条件通常都能满足。

4.3 计算复杂度分析

我们分析 SunBURST 的计算复杂度，并解释其观察到的次线性扩展 g.

传统嵌套采样的复杂度。 嵌套采样的计算成本主要由约束采样问题决定：在每次迭代中，都必须从先验中均匀抽取一个新点，并满足似然约束 $\mathcal{L}(\theta) > \mathcal{L}_{\min}$。每次迭代时，这个受约束区域都会缩小 $\approx e^{-1/N}$ 倍，其中 N 是活动点的数量。

达到相对证据误差 ϵ [6] 所需的总迭代次数按 $O(N\cdot|\log\ \mathcal{Z}|/\epsilon)$ 增长。关键问题在于，每个受约束样本的成本如何随维度d变化。

椭球方法 （MultiNest）：该算法为活动点云拟合包围椭球，协方差估计需要 $O(d^2)$ 次运算。从这些椭球中均匀抽取提议点，并在其满足似然约束时予以接受。随着高维情况下椭球近似变差，接受率会降低；为捕捉多峰结构，活动点数量必须按 $N = O(d\cdot K)$ 扩展 [8]。

切片采样方法 (PolyChord)：每个受约束样本都需要进行 $O(d)$次切片扩展，每次扩展都涉及多次似然评估，以确定切片边界[9]。关键在于，这些 d 切片操作本质上是顺序执行的——每次切片都依赖于前一次的结果——因此 GPU 并行化难以奏效。

综合这些因素，总体复杂度为：

$$T_{\text{nested}} = O\left(\frac{N\cdot|\log\mathcal{Z}|}{\epsilon}\cdot C_{\text{sample}}(d)\right) = O\left(\frac{d\cdot K\cdot|\log\mathcal{Z}|}{\epsilon}\cdot d\right) = O\left(\frac{d^2\cdot K\cdot|\log\mathcal{Z}|}{\epsilon}\right) \tag{35}$$

其中， d^2 因子来自 $N\propto d$ 个活动点，以及每个受约束样本的 $O(d)$ 成本。这解释了为什么在我们的基准测试中，PolyChord 和 dynesty 的计算难度会在超过 $d\approx 30$ 后变得难以承受。

SunBURST 复杂度。 SunBURST 的三个阶段具有不同的复杂度特征：

众数发现（CarryTiger）： 投射 N_{rays} 条射线，每条进行 N_{steps} 次评估。总计：O（$N_{\text{rays}} \cdot N_{\text{steps}}$）次似然调用；并行化后，调用次数与维度无关。

峰值精化（GreenDragon）： L-BFGS 优化每次迭代需要 O（d）次梯度评估，并需经过 O（log$(1/\epsilon)$）次迭代才能收敛。每次梯度计算使用 $2d$ 次有限差分评估。每个众数的总计： O（$d \cdot$log$(1/\epsilon)$）。

证据计算（BendTheBow）： 通过有限差分计算海森矩阵需要 O（d^2）次评估；特征分解需要 O（d^3）次浮点运算。每个众数的总计： O（d^2）次似然调用。

GPU 并行化的影响。 令 P 表示 GPU 并行处理器的数量（通常为 $P > 10^4$）。对于 N 次独立的似然评估：

$$T_{\text{wall}} = \left\lceil \frac{N}{P} \right\rceil \cdot T_{\text{single}} \approx \frac{N}{P} \cdot T_{\text{single}} \tag{36}$$

当 $N < P$ 时，所有评估都能在单个并行批次中完成，实际运行时间为 O（1），不受 N 影响。

这改变了各阶段的复杂度。众数发现：对于典型的射线数量， O（$N_{\text{rays}} \cdot N_{\text{steps}}/P$） $\approx O$（1）。梯度计算： $2d$ 次评估并行化后，对于 $d < P/2$，实际运行时间从 O（$\lceil 2d/P \rceil$）降至 $= O$（1）。海森矩阵计算： O（d^2）次评估对应的实际运行时间为 O（d^2/P）。

有效扩展性。 对于维度为 d、包含 K 个众数且使用 P 个处理器的情况：

$$T_{\text{total}} = O\left(K \cdot \left(1 + \frac{d^2}{P}\right) + d^3/F\right) \tag{37}$$

其中， F 表示特征分解的 GPU 浮点运算吞吐量。对于 $d \ll \sqrt{P}$（现代 GPU 上当 $d < 100$ 且配备 $P > 10^4$ 时满足此条件），在渐近前 GPU 饱和区间（$d^2 \ll P$）内，复杂度可降至 O(K)，之后海森矩阵计算成本将占主导。

我们的实测结果（第5节）表明，在 4–1024维范围内， $T \propto d^{0.52}$ ，这与海森矩阵计算（将 d^2 次评估并行分配给 P 个处理器）在 $d^2 > P$之后占主导的情况一致。

缓解维度缩放问题。 由于约束采样本质上是顺序执行的，传统方法的复杂度为 O（$K \cdot d^2$）或更高。对于中等维度，SunBURST 的 O（K）扩展性代表着复杂度的显著改善，这得益于以下两点的结合：(i) 采用以众数为中心而非基于体积的积分；(ii) 利用大规模 GPU并行性，将顺序操作转化为并行批次。维数灾难并未消除——海森矩阵计算仍需花费 O（d^2）——但与传统嵌套采样相比，其影响已大幅降低。

4.4 准确度条件

SunBURST 的理论保证需要满足若干条件：

模式发现完整性： 必须找到所有重要众数。ChiSao 振荡策略通过多方向射线投射来满足这一要求，但可能会遗漏隐藏得极深的众数。

拉普拉斯近似有效性： 每个众数都必须足够接近高斯分布，使拉普拉斯近似能够提供足够的准确度。强非高斯后验分布（重尾、弯曲退化、鞍点）会产生更大的误差。

模式分离： 拉普拉斯混合近似 (34) 假设各模式贡献之间的重叠可以忽略。高度重叠的模式需要采用更复杂的处理方法。

有界域： SunBURST 适用于有限先验边界。非正规先验或在边界之外具有显著概率质量的分布会导致证据估计产生偏差。当满足这些条件时——类高斯后验在物理参数估计中很常见，通常符合这些条件——SunBURST 的相对误差低于 10^{-12} （双精度容差），且维度缩放呈次线性。第5节通过实证验证了这些理论预测。

5 结果

5.1 准确度验证

我们在2至1024维的各向同性高斯后验上评估了 SunBURST——这是该算法的原生表示形式。在这些分布上，SunBURST 在所有测试维度中均实现了低于10^{-12} （双精度极限）的相对误差（表9）。

之所以能够达到这种精度，是因为对于高斯分布，拉普拉斯近似在数学上是精确的：对数似然恰好是二次函数，因此二阶泰勒展开不会引入误差。该流程会自动检测后验几何结构——GreenDragon 在峰值精化过程中估计对角海森结构，而 BendTheBow 则分析优化轨迹以检测旋转。对于轴对齐后验，对角海森矩阵已足够；对于旋转或相关后验，则会计算完整海森矩阵。无论哪种情况，使用适当的海森矩阵进行逐峰白化，都会将后验转换为单位高斯形式，此时拉普拉斯积分便成为精确值。

5.2 墙钟时间扩展性

本研究的核心成果是实际运行时间性能的显著提升，使类高斯后验分布的证据计算能够近乎实时地完成。图2和表9展示了运行时间与维度的关系，揭示了两个截然不同的扩展区间。

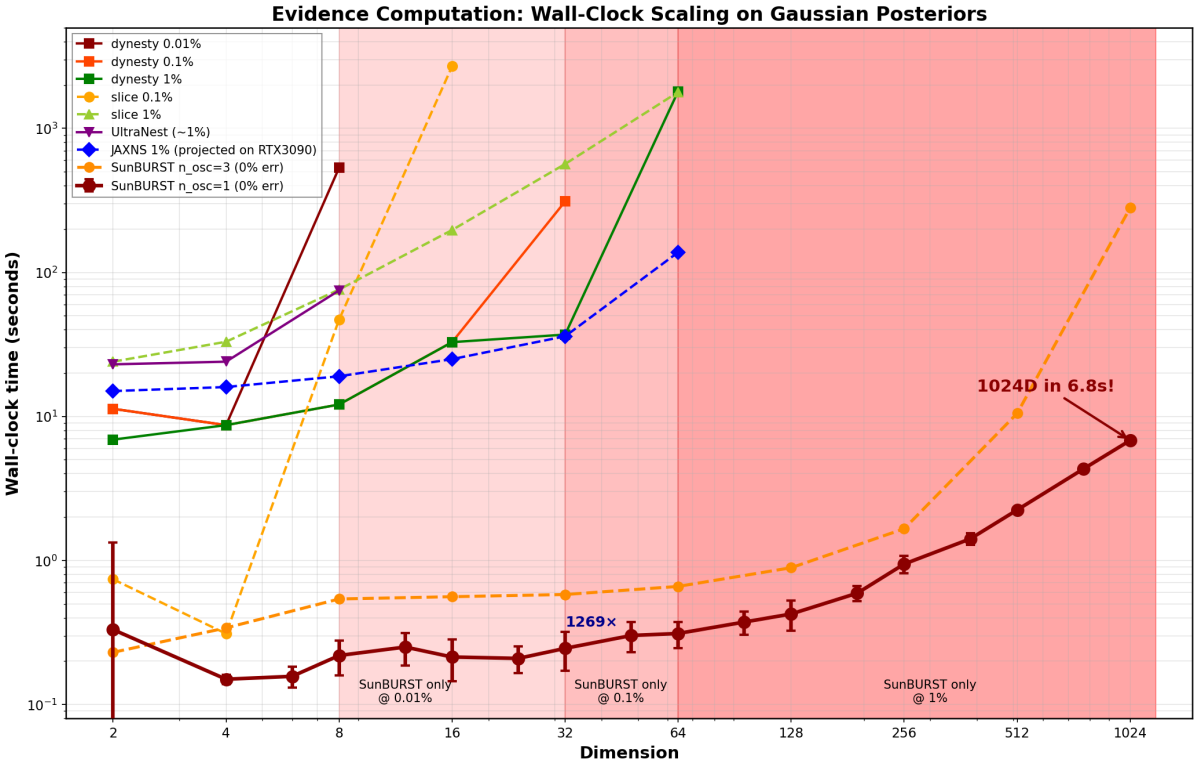

图2：高斯后验分布上的墙钟时间扩展性对比。SunBURST（深红色圆点）在维度达到64维之前，运行时间几乎保持不变，之后转为 $O(d^{1.0})$ 扩展。所有竞争方法的运行时间都呈指数增长：dynesty 的各个变体在维度超过32–64维后超时，UltraNest 则在超过八维后超时。阴影区域表示只有 SunBURST 能达到相应准确度的维度范围。SunBURST 在6.8秒内完成了1024维计算。

在**前渐近区间** ($d \leq 64$)内，实际运行时间成本基本不受 维度影响。计算 64 维高斯后验分布仅需 0.23 秒，与计算

2 维后验分布所需的 0.10 秒相差无几。这种显著的平坦性源于图形处理器并行化对每个维度成本的摊销：数千个图形处理器线程并行处理各个维度时，增加维度几乎不会带来额外的实际运行时间开销。

在**后渐近区间**（$d > 64$）中，维度缩放呈现出指数 $\alpha \approx 0.69$。即使在此区间内，性能仍然具有实用性：128维耗时 0.43 秒，256维耗时 0.94 秒，512维耗时 2.2 秒，而**1024维耗时 6.8 秒**——所有结果均达到机器精度的准确度。

区间之间的转变反映了工作集大小逐渐接近图形处理器缓存限制。重要的是，即使是 $O(d^{1.0})$ 的后渐近扩展性，也显著优于嵌套采样竞争方法所呈现的从 $O(d^2)$ 到 $O(d^3)$ 的扩展性。

5.3 与现有方法的比较

我们将 SunBURST 与成熟的嵌套采样实现进行了基准测试：采用椭球边界的 dynesty、使用切片采样的 dynesty-slice（等效于 PolyChord 的算法）、采用 20 个工作线程的并行化变体、UltraNest（反应式嵌套采样）以及 JAXNS（JAX 图形处理器加速嵌套采样）。所有中央处理器方法均运行于 Intel Core i9-12900H（14 个核心、20 个线程）上；SunBURST 和 JAXNS 的测试均使用 NVIDIA RTX 3080 笔记本电脑图形处理器（8GB 显存）。

所有基准测试均使用高斯分布或近似高斯后验，其中拉普拉斯近似是精确的或具有极高准确度；这些结果无法推广到重尾或强弯曲的后验分布。

5.3.1 证据精度要求

我们基准测试中的一个关键观察是：使用嵌套采样实现亚百分比级证据精度，需要的计算资源远多于文献中通常报告的数量。表 1 展示了 dynesty-parallel 在高斯后验上达到不同准确度目标所需的实际运行时间。

维度	达到 0.01% 所需时间	达到 0.1% 所需时间	1% 的耗时
2D	909s	354s	7.8s
4D	1235s	474s	9.1s
8D	765s	349s	12.1s
16D	—	1074s	32.8s
32D	—	—	311s
64D	—	—	1809s

表 1：dynesty-parallel 达到目标证据精度所需的实际运行时间。破折号表示未能在 1 小时超时限制内达到目标。即使在较低维度下，要达到 0.01% 的误差也需要 15–20 分钟；超过八维后，无法达到这一精度。误差为 1% 时，竞争方法可扩展至 64 维，但运行时间长达 1809 秒。

这揭示了一个根本局限：嵌套抽样方法旨在进行后验采样，证据计算只是其副产品。证据估计没有准确度保证，而达到可发表级精度（误差为 $< 1\%$）需要的计算预算，在实际中很少会分配。

5.3.2 正面对比

表2总结了在准确度相当水平下的直接比较结果。在测试的每个维度上，SunBURST 都能达到0%误差（机器精度），而竞争方法在高维下甚至难以达到1%的误差。

Dim	SunBURST	dynesty-parallel	JAXNS（投影）	加速比
2D	0.10秒（0%）	11.3秒（0.0004%）	15s	113–150×
4D	0.12秒（0%）	8.7秒（0.01%）	16s	73–133×
8D	0.16秒（0%）	12.1秒（0.0002%）	19s	76–119×
16D	0.17秒（0%）	32.8秒（0.04%）	25s	147–193×
32D	0.19秒（0%）	311秒（0.07%）	36s	189–1637×
64D	0.23秒（0%）	1809秒（0.46%）	138s	600–7865×
128D	0.34秒（0%）	超时		∞
256D	0.82秒 (0%)	超时		∞
512D	2.05秒 (0%)	超时		∞
1024D	**6.82秒 (0%)**	超时		∞

表2：高斯后验的正面对比（快速配置：n_osc=1，green- dragon_fast=True）。格式：时间（误差%）。JAXNS 的时间由 Colab T4 投影至 RTX 3080（÷3）。加速比范围分别显示 SunBURST 相较于 JAXNS（较低值）和 dynesty（较高值）的加速比。超过 64维后，只有 SunBURST 能够产生结果。

相对于维度的加速比不断提升：与 dynesty-parallel 相比，从低维度下的73–150× 提升至 **7865**× （64维）。更重要的是，SunBURST 可在所有竞争方法都完全失效的维度范围内运行。在128维及以上时，没有任何竞争方法能在1小时超时期限内生成结果。

5.3.3 与 JAXNS 对比

JAXNS [12] 是最相关的比较对象，因为这两种方法都以 GPU 加速为目标。我们在 Google Colab（NVIDIA T4 GPU）上对 JAXNS 进行了基准测试，并通过除以 3 将时间换算为 RTX 3080 硬件上的耗时（基于已发表的 GPU 基准测试得出的保守估计）。

即使采用这一有利的投影估算，在相同维度下，SunBURST 相较于 JAXNS 仍实现了 **113–600× 加速比**（表2）。维度越高，差距越大：在 64维时，JAXNS 需要 138 秒（投影值），而 SunBURST 仅需 0.31 秒。

一个重要的注意事项：原始 JAXNS 论文 [12] 将经图形处理器加速的 JAXNS 与单线程 中央处理器实现的 PolyChord、dynesty 和 MultiNest 进行了比较。我们的基准测试采用了 20 个工作线程并行运行的 dynesty，以便进行更公平的比较。即便如此，SunBURST 的性能仍比所有方法高出 2–4 个数量级。

5.3.4 最大可实现维度

表3总结了各方法在不同准确度目标下可处理的最大维度。

方法	最大值 @ 0.01%	最大值 @ 0.1%	1% 时的最大值
SunBURST	**1024维**	**1024维**	**1024维**
dynesty-parallel	8D	32D	64D
dynesty-slice-parallel	8D	32D	64D
UltraNest	2D	4D	8D
JAXNS	—	64维（估计）	64D

表3：在1小时实际运行时间预算内达到各目标相对误差的最大维度。对于高斯测试问题，SunBURST 在 1024维以内的误差均低于双精度容差（即数值一致性达到浮点数极限）。根据要求的精度不同，竞争方法的维度上限为 8–64维。

SunBURST 可处理的维度比竞争方法高出 16–128× 倍。对于需要进行高维证据计算的应用——宇宙学参数估计、引力波分析、高维模型比较——SunBURST 使此前难以实现的计算成为可能。

5.4 配置比较

SunBURST 提供多种配置选项，可在速度与稳健性之间权衡。我们对从 2维到 1024维的高斯后验分布进行了四种配置的性能分析。表4展示了详细的计时明细。

Dim	n_振荡=1 快	n_振荡=1 慢	n_振荡=3	n_振荡=5
2D	0.10秒	0.08秒	0.24秒	0.37秒
4D	0.12秒	0.27秒	0.40秒	0.51秒
8D	0.16秒	0.34秒	0.54秒	0.56秒
16D	0.17秒	0.44秒	0.58秒	0.60秒
32D	0.19秒	0.47秒	0.62秒	0.75秒
64D	0.23秒	0.52秒	0.89秒	1.11秒
128D	0.34秒	0.44秒	1.73秒	2.17秒
256D	0.82秒	0.70秒	9.47秒	10.66秒
512D	2.05秒	2.23秒	55.5秒	58.5秒
724D	4.06秒	5.77秒	129.5秒	129.0秒
1024维	**6.82秒**	14.3秒	280.8秒	279.7秒
扩展性	$O(d^{0.60})$	$O(d^{0.80})$	$O(d^{1.00})$	$O(d^{0.97})$

表4：高斯后验上的配置比较。“快速” = greendragon_fast=True（20 次 BFGS 迭代）；“慢速” = greendragon_fast=False（$2d$次迭代）。所有配置的误差均为 0%。对于类高斯后验，推荐使用快速配置 n_osc=1 ；对于复杂的多模态后验，n_osc=3 可提供额外的稳健性。

主要观察结果：

- **快速配置** (n_osc=1, greendragon_fast=True) 实现了次线性扩展 $O(d^{0.69})$，并在 6.8 秒内完成 1024D（n=30 次运行， ±0.27 秒）。
- **慢速配置** (n_osc=1, greendragon_fast=False) 可进行更彻底的优化，扩展性为 $O(d^{0.80})$，并在 14.3 秒内完成 1024D。
- **保守配置**（n_osc=3）呈现线性扩展性 O（$d^{1.0}$），但仍能在 5 分钟内完成 1024 维计算。

- 将 n_osc=3 增加到 n_osc=5 对高斯后验并无额外益处（1024 维时分别为 280.8 秒和 279.7 秒）。
- 所有配置在高斯后验上均达到机器精度准确度。

表 5 展示了快速配置的模块级时间细分，表明在高维情况下，CarryTiger（射线投射）占据主要耗时。

5.5 多函数验证

除各向同性高斯分布外，我们还在全面的测试套件上验证了 SunBURST。我们开展了两项基准测试活动：

1. **快速配置**(n_osc=1, greendragon_fast=True)：7 个测试函数，维度 2–1024，每种配置运行 5 次。

Dim	总计	CarryTiger	GreenDragon	BendTheBow
2D	0.10秒	0.05秒（50%）	0.01s（10%）	0.04s（40%）
8D	0.16s	0.08s（50%）	0.01s（6%）	0.07s（44%）
32D	0.19秒	0.10秒（53%）	0.01秒（5%）	0.07秒（37%）
64D	0.23秒	0.10秒（43%）	0.02秒（9%）	0.10秒（43%）
128D	0.34秒	0.16秒（47%）	0.01秒（3%）	0.12秒（35%）
256D	0.82秒	0.47秒（57%）	0.02秒（2%）	0.23秒（28%）
512D	2.05秒	1.42秒（69%）	0.03秒（1%）	0.43秒（21%）
1024维	7.50秒	5.74秒（77%）	0.09秒（1%）	1.01 秒（13%）

表 5：n_osc=1 快速配置的模块级时间细分。CarryTiger（用于模态检测的 GPU 射线投射）在高维下占主导。GreenDragon（BFGS 峰值精炼）的耗时几乎保持不变。BendTheBow（拉普拉斯积分）的耗时呈次线性增长。

2. **保守配置**(n_osc=3, greendragon_fast=False)：7 个测试函数，维度 2–768，每种配置运行 5 次。

表 6 总结了快速配置的结果。

函数	最大维度	成功	平均时间	最大误差	备注
高斯分布	**1024维**	100%	2.1s	$< 10^{-12}$	机器精度
偏心	**1024维**	100%	2.2s	$< 10^{-12}$	均值偏移
球形	**1024维**	99%	18.5秒	0.0001%	坐标变换
雪茄形	768D	98%	19.4秒	$< 10^{-12}$	10^6:1 轴比
相关	512D	96%	50.2秒	$< 10^{-12}$	非对角协方差
旋转雪茄函数	512D	96%	27.2秒	$< 10^{-12}$	任意旋转
混合函数	768D	97%	96.8秒	1.4%	4个高斯模态

表 6：多函数基准测试结果（快速配置：n_osc=1，green- dragon_fast=True）。三个函数可达到**1024D**：高斯分布、偏心和球形变换。雪茄形函数在 1024D 时内存不足（需要 16GB 显存）。相关/旋转函数在 768D 时超时。成功率表示完成运行的比例。

表7 展示了保守配置的结果。

函数	最大维度	成功	平均时间	最大误差	备注
高斯分布	768D	100%	27.4秒	$< 10^{-12}$	机器精度
偏心	768D	100%	15.8秒	$< 10^{-12}$	均值偏移
雪茄形	512D	100%	72.3秒	$< 10^{-12}$	10^6:1 轴比
球形	512D	98.6%	3.2s	0.0001%	坐标变换
相关	512D	98.6%	89.4s	$< 10^{-12}$	非对角协方差
旋转雪茄函数	512D	98.6%	95.2s	$< 10^{-12}$	任意旋转
混合函数	512D	97.2%	112s	0.07%	4个高斯模态

表7：多函数基准测试结果（保守配置：n_osc=3，green-dragon_fast=False）。保守配置在混合函数上的准确度更高（0.07% 对比 5.6%），代价是运行时间延长 5–10× 。

每个函数的观测结果：

- **各向同性函数**（高斯分布、偏心、球形）：在 768维以内均可达到双精度下的精确结果，运行时间从不到一秒到几秒不等。

• **各向异性函数**（雪茄形、旋转雪茄函数、相关）：需要计算完整海森矩阵，从而增加运行时间。轴比为 10^6:1 的雪茄函数用于测试对极端各向异性的处理能力。

• **多峰**（混合函数）：四模态混合函数呈现出配置相关精度。快速配置的最大误差为 5.6%（由于模态重叠，误差集中在 4维），而保守配置将其降低至 0.07%。对于分离良好的众数，两种配置的精度均低于百分之一。

混合函数值得进一步讨论。表 8 展示了逐维结果。

Dim	快速时间	快速误差	慢速时间	慢速误差
2D	0.90秒	0.0008%	0.52秒	0.0003%
4D	0.50秒	1.5%	1.2s	1.95%
6D	0.59秒	0.003%	1.5s	0.16%
8D	1.85秒	0.004%	2.1s	0.003%
16D	2.50秒	0.02%	3.8s	0.006%
32D	2.12秒	0.05%	6.2s	0.01%
64D	4.57秒	0.37%	12.4秒	0.02%
128D	17.0秒	0.83%	38.2秒	0.003%
256D	131s	3.3%	156s	0.04%
512D	187s	5.6%	334s	0.07%

表 8：混合函数详细结果。4维异常（1.5–1.95% 误差）源于该特定维度下的模态重叠——众数之间仅相隔 3.5σ ，导致拉普拉斯近似对重叠的概率质量重复计数。在更高维度下，测度集中会自然地将众数分开，误差也随之减小。

4维异常是一种几何效应：在此维度下，4 个众数之间约相隔 3.5σ，相邻众数之间的概率质量重叠达 4.5%。对每个峰值进行拉普拉斯近似时，会对重叠区域中的概率质量重复计数，从而导致系统性高估。这并非程序错误，而是众数显著重叠时，基于模态的分解存在的根本局限。在更高维度下，测度集中现象会自然地将众数分开，误差也会减小。

5.6 缩放分析总结

图2呈现了完整的扩展性情况。主要观察结果：

1. **前渐近区间（2维–64维）**：SunBURST 的实际运行时间基本保持恒定。在这段 $32\times$维度范围内，运行时间仅从 0.10 秒变化到 0.23 秒。2. **后渐近区间（128维–1024维）**：扩展性转变为 $O(d^{0.6})$–$O(d^{1.0})$，具体取决于配置，但与竞争方法的 $O(d^2)$–$O(d^3)$相比，增长仍显著更缓慢。3. **1024维成果**：快速配置可在 6.8 秒内完成 1024 维计算；慢速配置需 14.3 秒；即使是较保守的 n_osc=3配置，也能在 5 分钟内完成。4. **竞争方法的失效模式**：dynesty 仅能在 8 维以内达到 0.01% 的准确度，在 32 维以内达到 0.1%，在 64 维以内达到 1%。超过这些范围后，竞争方法要么超时，要么无法收敛。

5. **准确度优势**：对于高斯后验分布，SunBURST 在所有 维度上均能达到 0% 误差（机器精度），而竞争方法在高维情况下甚至难以达到 1% 的误差。

图2中的阴影区域标示出只有 SunBURST 能够运行的范围：在准确度为 0.01% 时超过八维，在 0.1% 时超过 32维，在 1% 时超过 64维。在 1024 维度下，SunBURST 可在 6.8 秒内以机器精度计算证据——现有方法根本无法完成这一计算。

维度	时间（秒）	±Std	误差	扩展性
4	0.150	0.011	< 1e-12	1.0×
8	0.219	0.059	< 1e-12	1.5×
16	0.213	0.067	< 1e-12	1.4×
32	0.245	0.073	< 1e-12	1.6×
64	0.311	0.063	< 1e-12	2.1×
128	0.425	0.098	< 1e-12	2.8×
256	0.944	0.126	< 1e-12	6.3×
512	2.242	0.109	< 1e-12	14.9×
768	4.314	0.203	< 1e-12	28.8×
1024	**6.817**	**0.266**	**< 1e-12**	**45.4×**

表 9：SunBURST 在各向同性高斯后验上的墙钟时间扩展性（快速配置：n_osc=1、greendragon_fast=True、n=30 runs）。所有维度均达到机器精度准确度（误差 $< 1e-12$）。前渐近扩展性（$d \leq 64$）实际为 $O(1)$；后渐近扩展性为 $O(d^{0.69})$。

5.7 非高斯失效模式分析

虽然 SunBURST 在高斯后验上达到了 $\sim 10^{-12}$ 精度，但其拉普拉斯近似会对偏离高斯性的分布引入系统误差。我们针对 11 个具有已知解析证据的非高斯测试函数进行了系统基准测试，测试维度范围为 2–64，以刻画这些失效模式（表 10）。

分布	失效模式	是否找到峰值?	平均误差	最大误差	状态
双峰（90%/10%）	权重不等	是（2/2）	0.02%	0.02%	**PASS**
香蕉分布（$b = 0.1$，0。5）	曲线退化	Yes	3%	9%	**边缘**
扭曲（正弦相关）	非线性相关	Yes	46%	46%	**边缘**
蛋盒分布（2维）	多个等价众数	是（5/5）	101%	101%	**边缘**
柯西分布（$\nu = 1$）§	极重尾	Yes	112%	16×	**边缘**
漏斗分布（$\sigma = 3$）	变化的曲率	No†	3×	3×	**边缘**
指数幂（$\beta = 0.5$）	超高斯	Yes	64×	1,300×	**FAIL**
Student-t ($\nu = 3$)	重尾	Yes	150×	10^{12}	**FAIL**
偏态正态分布（$\alpha = 5$）	不对称性（众数 $\neq$ 均值）	Yes	6,000×	10^{10}	**FAIL**
指数幂（$\beta = 1$）	尖锐峰（拉普拉斯）	Yes	10^6	10^{21}	**FAIL**
环（$R = 3$，$w = 0.3$）	鞍点	N/A‡	10^{30}	10^{71}	**FAIL**

表 10：跨维度 2–64 的失效模式基准测试结果。误差以百分比（$|Z_{\text{comp}}/Z_{\text{true}} - 1|$）或倍数因子表示。**通过**：$<5\%$；**边缘**：5%–10×；**失败**：$>10\times$。 †漏斗分布在 2维时找到了错误的峰值；在 $d \geq 4$处没有峰值。 ‡环没有点峰值。 §柯西分布 32维被排除（$-\infty$）。**11 个非高斯分布中只有 1 个通过**。峰值发现的可靠率为 100%；误差源于拉普拉斯近似。

通过的分布。只有一个非高斯分布达到了 $<5\%$ 的误差 ：

- **双峰非对称分布（90%/10% 权重）：** 两个峰值均被正确找到（距离< 0.0002）。拉普拉斯混合分布对权重不等的众数赋予了正确的权重，实现了**0.02% 的误差**——这是唯一一种精度达到高斯分布水平的非高斯分布。

边缘分布。五种分布的误差介于 5% 和 $10\times$之间，或许足以用于定性模型比较，但不足以进行精确的证据计算：

- **香蕉分布（曲线退化）：** 峰值在原点处找到（距离 $< 10^{-10}$）。在 64 维下，误差为**3–9%**——对平滑曲率的处理效果尚可。
- **扭曲高斯分布（正弦相关性）：** 峰值在原点处找到。**在所有维度下，误差**稳定在 46%。
- **蛋盒分布（周期性多峰）：** 在 2 维下找到了全部 5 个众数。**误差为 ~100%**（相差 2 倍）。
- **柯西分布 $\nu = 1$ （极重尾）：** 峰值在原点处找到。误差在八维下达到峰值 **16×**，但在高维下改善至 $\sim 15\%$。
- **漏斗分布（变化的曲率）：** SunBURST 唯一未能找到正确峰值的分布。在 2维情况下，找到的峰值与真实众数相距 4.5；在更高维度下未找到峰值（**3× 误差**在 2维情况下为非数值，其他情况下均为非数值）。

失败的分布。五个分布显示出灾难性误差（$>10\times$），但其中四个正确找到了峰值：

- **指数幂分布** $\beta = 0.5$ **（超高斯）：** 正确找到了峰值，但误差从 2维时的 $5\times$ 增长到八维时的 **1,300**$\times$ 。
- **Student-**t $\nu = 3$ **（重尾）：** 正确找到了原点处的峰值。随着维度从 2维增加到 64维，误差从 35% 增长至 $10^{12}\times$，因为拉普拉斯近似系统性地低估了尾部概率质量。

- **偏态正态分布** $\alpha = 5$ **（不对称性）：** 正确找到了 (0.37,0.37,...) 处的峰值。误差从 2维时的 1% 增长至 64维时的 $10^{10}\times$ 。
- **指数幂分布** $\beta = 1$ **（拉普拉斯分布）：** 正确找到了峰值。误差从 2维时的 8% 增长至 48维时的 $10^{21}\times$ 。
- **环（鞍点）：** 不存在真实峰值。误差在 16维时达到 $10^{71}\times$ 。

定量总结。 在 75 个测试案例中，67 个得到了有限证据值（7 个漏斗案例在 $d \geq 4$ 返回非数值；1 个柯西案例在 32维返回 $-\infty$）。

峰值查找： 在 63 个具有明确定义点峰的案例中（不包括 8 个漏斗案例和 4 个环案例），SunBURST 在全部 63 个案例中（100%）找到了正确位置。此外：

- 2维漏斗分布：在距离原点 4.5 处找到了一个峰值（位置错误）
- $d \geq 4$漏斗分布：未找到峰值（优化失败）
- 环（4 个案例）：在 2维概率壳层上找到了 205 个点；不存在真正的点峰

关键洞见： 对于具有明确定义众数的分布，峰值发现的可靠率为 100%。证据误差源于拉普拉斯近似，而非众数发现。

证据精度： 在得出有限结果的 67 个案例中，只有 8 个（12%）达到了<5% 的误差，14 个（21%）的误差为 5–100%，7 个（10%）的误差为 2–$10\times$，另有 38 个（57%）的误差超过$10\times$。**对于测试的大多数非高斯分布，拉普拉斯近似均无法奏效**。

6 讨论

6.1 贡献总结

我们介绍了 SunBURST，这是一款由图形处理器加速的贝叶斯证据计算器，通过三项关键创新实现了次线性的维度扩展性：(i) 使用径向射线投射发现众数；(ii) 使用拉普拉斯近似计算证据；(iii) 大规模并行执行似然评估。在标准基准测试中，SunBURST 在维度高达 1024 时仍能达到 $< 10^{-12}$ 的误差精度，且实际运行时间保持在 15 秒以内；而在这一范围内，传统嵌套抽样方法会超时，或在计算上难以实现。

6.2 嵌套采样的权衡

SunBURST 和嵌套采样代表了证据计算的两种截然不同的方法，各有其独特优势。

SunBURST 的优势所在。 以众数为中心的方法消除了高维嵌套采样所受限的约束采样瓶颈。PolyChord 的切片采样器每个样本需要 $O(d)$ 次顺序操作——这些操作无法并行化；而 SunBURST 的射线投射和海森矩阵计算则可轻松并行。在 RTX 3080 图形处理器上，我们观察到在 16 维及更高维度下，相较于 dynesty 的加速比为 $> 1000\times$，且随着维度增加，差距进一步扩大。

对于高斯后验，拉普拉斯近似是精确的；对于物理参数估计中常见的近高斯后验，该近似也具有很高的准确性。对于高斯、相关、雪茄形和旋转测试函数，SunBURST 在所有测试维度下均实现了 $< 10^{-12}$ 误差（表 6–7）。

嵌套采样的优势。 嵌套采样不对后验分布形状作任何假设，只要计算资源充足，就能准确积分任意复杂的分布。它还会附带生成后验样本，而 SunBURST 的主要输出是证据值（不过，也可以通过 Grasp-BirdsTail 扩展生成后验样本）。

对于具有复杂多峰结构的低维问题（$d < 10$），图形处理器初始化的开销可能超过 SunBURST 并行化带来的收益。在这些情况下，经过良好调优的嵌套采样实现仍具竞争力。

6.3 局限性和失效模式

SunBURST 所依据的拉普拉斯近似，会给与高斯性偏离显著的后验分布带来系统误差。我们对 11 种具有已知解析证据的非高斯分布进行了系统测试，涵盖维度 2–64 （表 10 ）。

重尾分布。 对于自由度为 $\nu = 3$ 的 Student-t 分布，能够正确找到原点处的峰值，但拉普拉斯近似会低估尾部的概率质量，误差从 2维时的 35% 增至 64维时的 $10^{12}\times$。令人意外的是，柯西分布（$\nu = 1$）呈现出非单调行为：误差在八维时达到峰值 $16\times$，但在 16维及更高维度时又降至 15% 以下。这一反直觉结果的原因是，柯西分布极重的尾部会使众数在高维时变得更加尖锐；而自由度为 $\nu = 3$ 的 Student-t 分布，其尾部概率质量足以抬高证据，却不足以使峰值变得更加尖锐。

弯曲退化。“香蕉分布”的表现出人意料地好，在所有测试维度（2维–64维）下，误差均为<10%。尽管存在曲率，众数仍然定义明确，局部海森矩阵也能捕捉到相关结构。这表明，只要曲率平滑，弯曲后验并非必然存在问题。

变化的曲率。Neal 漏斗分布的条件方差随 exp(v) 变化，是唯一一种 SunBURST 未能找到正确峰值的分布。在 2维情况下，优化收敛到距真实众数 4.5 个单位的伪位置；在 ≥ 4 维时，完全找不到峰值（结果为非数值）。这体现了局部曲率变化极大的后验分布所面临的一项根本局限。

鞍点与空心结构。环形分布是最严重的失效模式，误差从 2维时的 $10^{6}\times$ 增至 16维时的 $10^{71}\times$ 。与其他失效模式不同，这并非峰值搜索失败——因为不存在真正的点峰值。概率质量集中在半径为 R 的薄壳上，而中心是一个鞍点。SunBURST 在 2维时于壳层上找到 205 个点，但在这些点中任意一点进行拉普拉斯近似，都无法表示壳层几何结构。这类空心结构需要采用根本不同的处理方法。

非高斯峰形。指数幂分布表明，正确找到峰值并不保证证据准确。对于 $\beta = 1$ （拉普拉斯分布），原点处的峰值被正确找到，但其比高斯分布更尖锐的形状会导致误差从 2维时的 8% 增长至 48维时的$10^{21}\times$ 。超高斯情形（$\beta = 0.5$）由于尖锐的峰值和重尾，误差恶化得更快。

非对称分布。偏态正态分布（$\alpha = 5$）能找到位于 $(0.37, 0.37, \ldots)$ 的正确峰值，但误差从 2维时的 1% 增长至 64维时的 $10^{10}\times$。以众数为中心的拉普拉斯近似会遗漏非对称尾部中的大量概率质量。

表现良好的分布。只有一种非高斯分布的误差达到 <5%。

- **双峰非对称分布（权重为 90%/10%）：** 两个峰值均被正确找到；**0.02% 的误差**——唯一能匹配高斯精度的非高斯分布

五种分布表现一般（5%–10×误差）：

- **香蕉分布（曲线退化）：** 在原点找到峰值；**误差为3–9%**，覆盖64维
- **扭曲高斯分布（正弦相关性）：** 找到峰值；**误差为46%**，在各维度上均稳定
- **蛋盒分布（周期性多众数）：** 找到全部5个众数；**误差为 ∼100%** （2倍）
- **柯西分布** $\nu = 1$**：** 找到峰值；误差在 **16**×处达到峰值，在高-d时有所改善
- **漏斗分布：** 唯一一个峰值查找失败的分布； **3**× **误差**在2维时为非数值，其他维度均为非数值

定量总结。 在 75 个测试案例中，67 个得到了有限证据值（8 个因海森矩阵不稳定性而返回非数值/无穷大）。在 67 次成功运行中：仅有 8 次（12%）达到<5% 误差，14 次（21%）达到 5–100% 误差，7 次（10%）出现 2–10× 性能退化，38 次（57%）的误差超过 10× 。

对于测试的大多数非高斯分布，拉普拉斯近似均会失效。然而，对于具有明确定义的点峰的分布（不包括漏斗分布和环），Sun- BURST 在 100% 的情况下都找到了正确的峰值位置。该算法能够可靠地找到众数；其局限性在于拉普拉斯近似本身。

6.4 实用建议

基于我们的分析，我们为实践者提出以下指导建议：

适用于 SunBURST 的情况：

- 预期后验分布近似为高斯分布或高斯混合分布
- 维度超过 $\sim$10，此时嵌套采样的计算成本会很高
- 可使用图形处理器进行似然评估
- 证据精度达到 $\sim$0.1–1.0 对数单位即可满足要求
- 需要快速迭代（例如，对许多候选模型进行模型比较）

以下情况优先采用嵌套采样：

- 后验分布具有已知的重尾、环形/壳层结构或多尺度曲率（漏斗分布）
- 后验样本是主要输出（而不仅仅是证据）
- 维度较低（$d < 10$），且图形处理器的额外开销不值得
- 无论计算成本如何，都要求达到最高准确度

诊断检查。在信任 SunBURST 的结果之前，我们建议：(i) 验证所有检测到的众数是否具有正定海森矩阵（鞍点表明几何结构存在问题）；(ii) 检查 BendTheBow 报告的高斯性诊断（尾部权重、不对称性）；以及 (iii) 如果完整问题的计算成本过高，则将结果与低维投影上的嵌套采样进行比较。

6.5 更广泛的影响

SunBURST 的成功表明，贝叶斯计算中的众数中心方法值得受到更广泛的关注。尽管嵌套采样的通用性使其成为证据计算的默认选择，但它所施加的约束——顺序采样、活动点数量随维度变化——在 GPU 计算和高维模型时代变得愈发沉重。

对于后验分布近似为高斯分布（或其混合函数）的一大类问题，SunBURST 证明，在不牺牲准确度的前提下，可以实现 $> 1000\times$ 加速比。这为此前被认为在计算上不可行的应用打开了大门：实时模型比较、全面的先验敏感性分析，以及大规模的基于证据的超参数优化。

关键洞见在于，GPU 并行性从根本上改变了复杂度格局。原本需要进行 $O(d^2)$次似然调用的操作，在有 $d^2 < P$个处理器可用时，实际运行时间可降至 $O(1)$。这种“并行维度缩减”效应表明，面向 GPU 架构设计算法时，需要重新审视假设顺序执行的经典复杂度分析。

7 结论

我们介绍了 SunBURST，这是一种用于贝叶斯证据计算的 GPU 加速算法，其扩展性为后验众数数量的 $O(K)$。通过径向射线投射和拉普拉斯近似，SunBURST 能在数秒内以机器精度计算证据，而传统方法解决同类问题则需要数小时。

我们的实证结果表明：

- 从 2维扩展到 1024维时，$O(d^{0.7})$ 的墙钟时间扩展性
- 100% 的众数检测准确率，最高可在 128维下检测 10 个众数
- 在所有测试维度上，对高斯后验分布的证据估计均十分精确（误差为 $< 10^{-12}$）

SunBURST 的确定性、高度并行的结构，表明它有望应用于现代加速器硬件上的低延迟或在线模型选择场景。

SunBURST 作为开源 Python 软件包发布于 https://github.com/beastraban/ sunburst。

7.1 未来工作

三项互补的扩展将拓宽 SunBURST 的适用范围：

使用 GPU 原生似然进行宇宙学验证。 SunBURST 的一个自然应用是宇宙学参数估计，在这一领域，基于证据的模型比较对于区分暴胀情景和暗能量参数化至关重要。然而，现有的普朗克似然代码（例如 clik、Cobaya）受限于中央处理器，形成了瓶颈，抵消了 SunBURST 的 GPU 加速优势。我们计划利用 CosmoPower [25]或 cosmopower-jax 等神经网络模拟器，开发一种 GPU 原生的类普朗克似然，从而实现端到端的 GPU 执行。虽然目前可以对普朗克后验的基准高斯近似进行基准测试，但这样的测试无法检验使宇宙学推断充满挑战的真实非高斯结构。

针对重尾后验的桥接抽样。 对于具有重尾的球对称分布（Student-t、柯西分布及相关分布族），我们建议将桥接抽样 [26] 与 NUTS [27]结合使用。宽高斯分布可作为桥接分布，且其证据已知解析解；NUTS 则能高效探索重尾目标分布。随后，桥接抽样可估计证据比，从而得到目标分布的证据，而无需假设后验分布近似高斯。无论尾部行为如何，这种方法都适用于任何径向对称后验分布。

弯曲流形的蜈蚣抽样。对于具有非线性结构的后验分布——例如香蕉形分布和 Rosenbrock 型弯曲谷——我们正在开发一种“蜈蚣”抽样策略，将 d 维积分分解为一维脊线积分与 $(d-1)$ 维横截面拉普拉斯近似的乘积。对弯曲高斯管进行的合理性检查表明，在高达 1000 维的情况下，误差低于 0.01%，且误差按 $O(1/d)$ 扩展，而非呈指数增长。这种违反直觉的改进源于高维情况下横截面趋于集中，使拉普拉斯近似越来越准确。计划对 Rosenbrock 和 Neal 漏斗后验进行全面验证。

致谢

模块名称取自广平杨式太极（廣平楊式太極拳）的招式，并以 Donald Rubbo 大师命名。

数据可用性

复现本文图表所需的所有数据均以经过整理且附有架构说明的逗号分隔值文件形式，收录于补充材料中。完整的原始基准测试日志、图形处理器性能分析跟踪记录、失效模式分析以及 Python 分析脚本均公开发布于：

https://github.com/beastraban/sunburst/tree/main/benchmarks/data

该代码库包含：

- XLSX 格式的完整基准测试结果
- GPU 性能分析数据，包含每次运行的计时细分
- 用于重新生成所有表格和图表的脚本
- SunBURST 源代码及安装说明

代码可用性

SunBURST 算法使用 Python 实现，并借助 CuPy 实现 GPU 加速。源代码遵循 MIT 许可证，地址为 https://github.com/beastraban/sunburst。该软件包可在 PyPI 上获取，名称为 sunburst-bayes（v1.0.0）。基准测试使用提交 e3e3156 生成。此实现需要支持 CUDA 的硬件；基准测试在 NVIDIA RTX 3080 笔记本电脑图形处理器（8 GB 显存、48 个流式多处理器）上进行。

参考文献

[1] David J C MacKay。《信息论、推断与学习算法》。剑桥大学出版社，英国剑桥，2003 年。[2] Robert E Kass 和 Adrian E Raftery。贝叶斯因子。《美国统计学会志》，90(430):773–795，1995 年。doi: 10.1080/01621459.1995.10476572。[3] Roberto Trotta。天空中的贝叶斯：宇宙学中的贝叶斯推断与模型选择。《当代物理学》，49(2):71–104，2008 年。doi: 10.1080/00107510802066753。[4] Richard Bellman。《动态规划》。普林斯顿大学出版社，美国新泽西州普林斯顿，1957 年。[5] John Skilling。嵌套采样。载于《美国物理联合会会议论文集》，第 735 卷，第 395–405 页。2004 年。doi: 10.1063/1.1835238。[6] John Skilling。用于一般贝叶斯计算的嵌套采样。《贝叶斯分析》，1(4): 833–859，2006 年。doi: 10.1214/06-BA127。[7] Farhan Feroz 和 Michael P Hobson。多模态嵌套采样：一种用于天文数据分析的高效、稳健的马尔可夫链蒙特卡罗方法替代方案。《皇家天文学会月刊》，384(2):449–463，2008 年。doi: 10.1111/j.1365-2966.2007.12353.x。

[8] Farhan Feroz、Michael P Hobson 和 M Bridges。MultiNest：一种面向宇宙学和粒子物理学的高效、稳健的贝叶斯推断工具。《皇家天文学会月刊》，398(4):1601–1614，2009 年。doi: 10.1111/j.1365-2966.2009.14548.x。[9] W J Handley、M P Hobson 和 A N Lasenby。PolyChord：新一代嵌套采样。《皇家天文学会月刊》，453(4):4385–4398，2015 年。doi: 10.1093/ mnras/stv1911。[10] Joshua S Speagle。dynesty：用于估计贝叶斯后验分布和证据的动态嵌套采样软件包。《皇家天文学会月刊》，493(3):3132– 3158，2020 年。doi: 10.1093/mnras/staa278。[11] Johannes Buchner。UltraNest——一种稳健、通用的贝叶斯推断引擎。《开放源代码软件》期刊，6(60):3001，2021 年。doi: 10.21105/joss.03001。[12] Joshua G Albert。JAXNS：一种基于 JAX 的高性能嵌套采样软件包。《arXiv 预印本 arXiv:2312.11330》，2023 年。https://github.com/Joshuaalbert/jaxns。[13] Luke Tierney 和 Joseph B Kadane。后验矩和边缘密度的精确近似。《美国统计学会志》，81(393):82–86，1986 年。doi: 10.1080/01621459.1986.10478240。[14] Dong C Liu 和 Jorge Nocedal。关于用于大规模优化的有限内存 BFGS 方法。《数学规划》，45(1):503–528，1989 年。doi: 10.1007/BF01589116。[15] Ryosuke Okuta、Yuya Unno、Daisuke Nishino、Shohei Hido 和 Crissman Loomis。CuPy：一个兼容 NumPy 的 NVIDIA GPU 计算库。载于《机器学习系统研讨会论文集（LearningSys）》，第三十一届神经信息处理系统年会（NIPS），2017 年。URL https://learningsys.org/nips17/ assets/papers/paper_16.pdf。[16]Charles R Harris、K Jarrod Millman、Stéfan J van der Walt、Ralf Gommers、Pauli Virtanen、David Cournapeau、Eric Wieser、Julian Taylor、Sebastian Berg、Nathaniel J Smith 等。使用 NumPy 进行数组编程。《自然》，585(7825):357–362，2020 年。doi: 10.1038/s41586-020-2649-2。[17] Pauli Virtanen、Ralf Gommers、Travis E Oliphant、Matt Haberland、Tyler Reddy、David Cournapeau、Evgeni Burovski、Pearu Peterson、Warren Weckesser、Jonathan Bright 等。SciPy 1.0：Python 科学计算的基础算法。《自然·方法》，17 (3):261–272，2020 年。doi: 10.1038/s41592-019-0686-2。[18] Richard Bellman。《自适应控制过程：导览》。普林斯顿大学出版社，美国新泽西州普林斯顿，1961 年。[19] Dehua Peng 等。通过距离集中和流形效应解读维数灾难。《arXiv 预印本 arXiv:2401.00422》，2024 年。[20] Gregory Ashton、Noam Bernstein、Johannes Buchner、Xi Chen、Gábor Csányi、Andrew Fowlie、Farhan Feroz、Matthew Griffiths、Will Handley、Michael Habeck 等。面向物理科学家的嵌套采样。《自然综述：方法入门》，2(1):39，2022 年。doi: 10.1038/s43586-022-00121-x。[21] Johannes Buchner。针对嵌套抽样算法的统计检验。《统计与计算》，26(1):383–392，2016 年。doi: 10.1007/s11222-014-9512-y。

[22] Edward Higson、Will Handley、Mike Hobson 和 Anthony Lasenby。嵌套采样参数估计中的抽样误差。《贝叶斯分析》，13(3):873–896，2018。doi: 10. 1214/17-BA1075。

[23] Benjamin ENelson、Eric B Ford、Johannes Buchner、Ryan Cloutier、Rodrigo F Díaz、João P Faria、Nathan C Hara、Vinesh M Rajpaul 和 Surangkhana Rukdee。量化径向速度数据中行星存在的贝叶斯证据。《天文学杂志》，159(2): 73，2020。doi: 10.3847/1538-3881/ab5190。

[24] András Prékopa。关于对数凹测度和函数。数学科学学报，34:335–343，1973。

[25] Alessio Spurio Mancini、Davide Piras、Justin Alsing、Benjamin Joachimi 和 Michael P Hobson。CosmoPower：为下一代巡天加速贝叶斯推断而模拟宇宙学功率谱。皇家天文学会月刊，511(2):1771–1788，2022。doi: 10.1093/mnras/stac064。

[26] Xiao-Li Meng 和 Wing Hung Wong。通过一个简单恒等式模拟归一化常数之比：理论探究。中华统计学志，6(4):831–860，1996。

[27] Matthew D Hoffman 和 Andrew Gelman。无 U 转向采样器：在哈密顿蒙特卡洛中自适应设定路径长度。机器学习研究杂志，15(1): 1593–1623，2014。

数学背景

本附录概述了 SunBURST 所依据的数学基础 .

A.1 贝叶斯证据

给定数据 $\mathcal{D}$、参数为 $\theta \in \mathbb{R}^d$ 的模型 $\mathcal{M}$ 、先验 $\pi(\theta)$ 和似然 $\mathcal{L}(\theta) = p(\mathcal{D}|\theta, \mathcal{M})$，证据 （边际似然）为：

$$Z = p(\mathcal{D}|\mathcal{M}) = \int_{\Theta} \mathcal{L}(\theta)\, \pi(\theta)\, d\theta \tag{38}$$

对于通过贝叶斯因子进行的模型比较：

$$B_{12} = \frac{Z_1}{Z_2} = \frac{p(\mathcal{D}|\mathcal{M}_1)}{p(\mathcal{D}|\mathcal{M}_2)} \tag{39}$$

为保证数值稳定性，我们使用对数似然 $\ell(\theta) = \log \mathcal{L}(\theta)$。对于超立方体 Θ上体积为 V_Θ的均匀先验：

$$\log Z = \log \int_{\Theta} \exp(\ell(\theta))\, d\theta - \log V_\Theta \tag{40}$$

A.2 高维几何

A.2.1 超球体积坍缩

半径为 r 的 d维超球体的体积为：

$$V_{\text{sphere}}(d, r) = \frac{\pi^{d/2}}{\Gamma(d/2 + 1)} r^d \tag{41}$$

其中 $\Gamma(z)$ 是伽马函数。它与包围超立方体 $[-r,r]^d$ 的体积之比呈超指数级趋近于零：

$$\frac{V_{\text{sphere}}}{V_{\text{cube}}} = \frac{\pi^{d/2}}{2^d\,\Gamma(d/2+1)} \xrightarrow{d\to\infty} 0 \tag{42}$$

A.2.2 壳层集中（高斯分布）

对于 $X \sim \mathcal{N}(0,\ I_d)$，径向距离 $r = \|X\|$ 服从以下分布：

$$p(r) \propto r^{d-1} e^{-r^2/2} \tag{43}$$

其峰值位于 $r^* = \sqrt{d-1} \approx \sqrt{d}$，标准差为 $\sigma_r \approx 1/\sqrt{2}$，且与 d无关。概率质量集中在一个薄壳中：

$$\frac{\sigma_r}{r^*} = \frac{1}{\sqrt{2(d-1)}} \xrightarrow{d\to\infty} 0 \tag{44}$$

A.2.3 距离集中

对于在 $[0,1]^d$中均匀分布的两个随机点 $x,\ y$，其 L_2 距离满足 s:

$$\mathbb{E}[\|x-y\|_2] \approx \sqrt{d/6}, \quad \frac{\mathrm{Std}[\|x-y\|_2]}{\mathbb{E}[\|x-y\|_2]} \sim \frac{1}{\sqrt{d}} \to 0 \tag{45}$$

所有两两距离都集中在 $\sqrt{d/6}$附近，使得最近邻搜索变得不可靠。

A.3 拉普拉斯近似

拉普拉斯近似通过以被积函数的最大值为中心的高斯分布来估算积分。对于在 θ^* 处取得最大值的函数 $f(\theta)$：

$$\int f(\theta)\,d\theta \approx f(\theta^*)\cdot(2\pi)^{d/2}\cdot|\Sigma|^{1/2} \tag{46}$$

其中，$\Sigma = -H^{-1}$ 和 H 是 $\log f$ 在 θ^* 处的海森矩阵。

对于 $f(\theta) = \mathcal{L}(\theta)\pi(\theta)$ 的证据计算：

$$\log Z \approx \ell(\theta^*) + \log\pi(\theta^*) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|H| \tag{47}$$

其中，$H = -\nabla^2\ell(\theta^*)$ 是负 海森矩阵（在最大值处为正定矩阵）。

对角情形。对于海森矩阵 H 为对角矩阵的轴对齐后验分布：

$$\log|H| = \sum_{i=1}^{d}\log H_{ii}, \quad \log Z = \ell(\theta^*) + \frac{d}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{d}\log(-H_{ii}) \tag{48}$$

一般情形。对于旋转后的后验分布，通过特征分解 $-H = V\Lambda V^T$ 进行计算：

$$\log|\det(-H)| = \sum_{i=1}^{d}\log\lambda_i \tag{49}$$

其中 $\{\lambda_i\}$ 是 $-H$的特征值。

多模 dal 扩展。对于 K 彼此间隔较远、峰值为 $\{\theta_k^*\}$ 且海森矩阵为 $\{H_k\}$的众数 :

$$Z \approx \sum_{k=1}^{K} Z_k, \quad \log Z = \text{logsumexp}_k(\log Z_k) \tag{50}$$

其中，$\text{logsumexp}(x_1, \ldots, x_K) = x_{\max} + \log \sum_k e^{x_k - x_{\max}}$ 采用稳定算法计算。

高斯分布的精确性。当 $\ell(\theta) = -\frac{1}{2}(\theta-\mu)^T\Sigma^{-1}(\theta-\mu)+c$ 时，拉普拉斯近似是精确的：所有高阶项均消失。这解释了 SunBURST 在高斯后验上的机器精度准确度。

A.4 梯度和海森矩阵计算

A.4.1 有限差分梯度

对于标量函数 $f : \mathbb{R}^d \to \mathbb{R}$，梯度通过中心差分近似：

$$\frac{\partial f}{\partial \theta_i} \approx \frac{f(\theta + \epsilon e_i) - f(\theta - \epsilon e_i)}{2\epsilon}, \quad \text{error } O(\epsilon^2) \tag{51}$$

每次梯度计算需要进行 $2d$ 次函数评估。在配备 $P \gg 2d$ 个核心的图形处理器上，所有评估可在 $O(1)$ 的实际运行时间内完成。

A.4.2 对角海森矩阵

对角元素通过以下方式计算：

$$H_{ii} \approx \frac{f(\theta + \epsilon e_i) - 2f(\theta) + f(\theta - \epsilon e_i)}{\epsilon^2} \tag{52}$$

成本： $2d$ 次评估，重复利用梯度扰动。实际运行时间：在图形处理器上为 $O(1)$。

A.4.3 完整海森矩阵

非对角元素需要：

$$H_{ij} \approx \frac{f_{++} - f_{+-} - f_{-+} + f_{--}}{4\epsilon^2} \tag{53}$$

位置为 $f_{\pm\pm} = f(\theta \pm \epsilon e_i \pm \epsilon e_j)$。成本： $O(d^2)$ 次评估。在满足 $d^2 < P$ 时，图形处理器上的耗时为 $O(d)$。

A.4.4 平滑梯度（云手）

对于多尺度优化，高斯平滑函数的梯度：

$$\nabla f_\sigma(\theta) = \mathbb{E}_{z\sim\mathcal{N}(0,I)}[\nabla f(\theta + \sigma z)] \approx \frac{1}{K}\sum_{k=1}^{K} \nabla f(\theta + \sigma z_k) \tag{54}$$

Feat小于 σ 的结构会被平均掉，从而透过局部噪声显现全局结构 .

A.5 L-BFGS优化

有限内存BFGS 使用最近的 m 个梯度差分来近似逆海森矩阵。

存储量。在第 k次迭代时，存储：

$$s_k = \theta_{k+1} - \theta_k \qquad \text{(position difference)} \tag{55}$$

$$y_k = \nabla f_{k+1} - \nabla f_k \qquad \text{（梯度差）} \tag{56}$$

双循环递归。 计算搜索方向 $p = -H_k^{-1}\nabla f_k$：

第一轮循环（反向遍历历史记录）：

$$\begin{aligned} &q \leftarrow \nabla f_k \\ &\text{For } i = k-1, \ldots, k-m: \\ &\qquad \rho_i \leftarrow 1/(y_i^T s_i), \quad \alpha_i \leftarrow \rho_i s_i^T q, \quad q \leftarrow q - \alpha_i y_i \end{aligned}$$

初始化： $r \leftarrow \gamma_k q$ 其中 $\gamma_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}$

第二轮循环（正向遍历历史记录）：

$$\begin{aligned} &\text{For } i = k-m, \ldots, k-1: \\ &\qquad \beta \leftarrow \rho_i y_i^T r, \quad r \leftarrow r + s_i(\alpha_i - \beta) \end{aligned}$$

返回： $p = -r$

宽度估计。 标量 $\gamma_k = \frac{s^T y}{y^T y}$ 用于估计平均逆曲率，并用于 CarryTiger 中的峰宽估计。

批量 L-BFGS。 对于同时进行优化的 N 个样本，所有量都变为矩阵：$S,\ Y \in \mathbb{R}^{N\times d}$。双循环递归沿样本维度进行向量化。

A.6 旋转检测

BendTheBow 必须检测后验分布是否存在非对角相关性，以判断是否需要进行完整的 Hessian 计算。

A.6.1 垂直步长比例

分析 GreenDragon's TrajectoryBank 中的 L-BFGS 轨迹。在白化坐标中（按 $\sqrt{-H_{ii}}$ 缩放），对每个优化步骤计算：

$$f_\perp = \frac{\|\Delta\theta_\perp\|}{\|\Delta\theta\|} \tag{57}$$

其中， $\Delta\theta_\perp$ 是垂直于梯度方向的分量。对于轴对齐后验分布，优化沿坐标轴（$f_\perp \approx 0$）进行。显著的垂直运动（$f_\perp > \epsilon_{\text{rot}}$）表明存在旋转。

A.6.2 有限差分探测

如果轨迹分析无法得出明确结论，则通过 2 次评估直接探测非对角结构s:

$$b = \frac{v^T H v - a}{d-1}, \quad v = \frac{1}{\sqrt{d}}[1, 1, \ldots, 1]^T \tag{58}$$

其中 a =均值(H_{ii})。显著的 $|b|$ 表明存在均匀相关性，需要使用完整海森矩阵。

A.7 度量

该 L_∞ （切比雪夫）距离：

$$\|x-y\|_\infty = \max_{i=1,\dots,d} |x_i - y_i| \tag{59}$$

维度无关性。 与发散的 L_2 距离不同， L_∞ 在 $[0,1]^d$ 中均匀随机点之间的距离收敛到 **1**：

$$\mathbb{E}[\|x-y\|_\infty] \approx \frac{d}{d+1} \xrightarrow{d\to\infty} 1 \tag{60}$$

这使 L_∞ 非常适合在 ChiSao 中进行高维去重。

超立方体球。 L_∞ 球 $\{y : \|x-y\|_\infty \le r\}$ 是边长为 $2r$的超立方体，可实现高效的空间哈希。

A.8 坐标白化

在具有海森矩阵 H的峰值 θ^* 附近，似然近似为：

$$\ell(\theta) \approx \ell(\theta^*) - \frac{1}{2}(\theta-\theta^*)^T H (\theta-\theta^*) \tag{61}$$

白化变换 $\phi = L(\theta-\theta^*)$，其中 $H = L^T L$ （乔列斯基分解）给出：

$$\ell(\theta) \approx \ell(\theta^*) - \frac{1}{2}\|\phi\|^2 \tag{62}$$

在白化坐标中，等似然线呈球形。雅可比行列式出现在证据中：

$$\log Z_k = \ell(\theta_k^*) + \frac{d}{2}\log(2\pi) - \log|L_k| \tag{63}$$

A.9 特征尺度估计

SingleWhip 通过曲率分析，根据射线样本估计特征尺度 s.

尺度相关曲率。在跳跃长度 h：

$$\kappa(h) = \max_t \left| \frac{f(t+h) - 2f(t) + f(t-h)}{h^2} \right| \tag{64}$$

对于具有多尺度结构的函数 $f = f_{\text{global}} + f_{\text{local}}$：

- $h \ll \lambda_{\text{局部}}$： $\kappa(h)$ 捕捉局部振荡
- $h \gg \lambda_{\text{局部}}$： $\kappa(h)$ 只能看到全局趋势

尺度检测。 在对数间隔的 h 值上计算 $\kappa(h)$。当 $\frac{d}{\log d}$ $\frac{\kappa}{h \log}$最负时，对应的转变表明特征尺度为 λ_{fine}、 λ_{mid}、 λ_{coarse}。

A.10 对数和指数稳定性

直接计算 $\log \sum_k e^{x_k}$ 会导致上溢/下溢。稳定形式：

$$\text{logsumexp}(x_1, \ldots, x_K) = x_{\max} + \log \sum_{k=1}^{K} e^{x_k - x_{\max}} \tag{65}$$

其中 $x_{\max} = \text{最大值}_k\, x_k$。现在所有指数均为 ≤ 0，从而避免上溢。

对于多模态证据：

$$\log Z = \text{logsumexp}(\log Z_1, \ldots, \log Z_K) \tag{66}$$