# quantenergy.tech 中文使用说明（手机查阅版）

2026-08-16 · Hongping Zhang · 中文 / Chinese-language site guide

**EcoCompute 一句话定位：**别的量化工具教你「怎么量化」，这个网站告诉你「**该不该**量化」。它用真实 GPU 功耗测量（NVML 直采，不是 TDP 估算）回答一个问题：量化到底省电还是更费电？

预印本：*Weight-Only Quantization Does Not Always Save Energy*（SSRN #6854700）。

> **先说清楚测量边界（本页所有数字都适用）：**我们采集的是 **GPU 芯片（package）功耗**，不是整机墙上功耗——不含电源损耗、CPU、内存、散热，也不换算 PUE 或 $CO_2e$；工况是 **单流解码、batch 1、256 token**。重复次数逐点标注：主数据集是 n = 2（CV < 2%），RTX 4090 深挖是**每个配置 n = 1**。详见 方法与局限。

## 0 · 30 秒上手

1. 想看别人测过的数据 → 点「已测模型」标签。
2. 想知道你自己的模型量化后能耗怎么变 → 点「你的模型」标签，拖一下滑块。
3. 想在给定的延迟／显存上限下找最优配置 → 点「优化（约束）」标签。
4. 想在自己的 GPU 上复现测量 → 点「自己跑」标签，一行命令。

## 1 · 顶部栏（Hero）

- 标题：**EcoCompute.**
- 核心标语：每个量化工具都告诉你怎么量化，我们告诉你**该不该**量化。
- 来源徽标：数据来自预印本，覆盖 4 种 NVIDIA GPU 架构（Turing / Ada / Ampere / Blackwell）的直接 NVML 功耗测量；Hopper 是借用 Ampere 曲线的外推，页面上会标出来。
- 两个按钮：「在 SSRN 读预印本」看论文方法与结论；「RTX 4090 实测数据集 · Zenodo DOI」下载我们自己在 RTX 4090 上跑出的实测数据。
- 右上角 About 链接 → 打开详细说明弹窗（见第 6 节）。

## 2 · 标签一：已测模型（Tested models）

功能：看「已经测过」的真实数据，判断某个 GPU + 模型上量化到底是省电还是费电。

- 两个下拉框：先选 GPU，再选模型。
- **卡片①·每 100 万 token 的推理能耗**：柱状图对比 FP16 / NF4 / INT8，下方给出结论（省还是赔），可一键「复制决策摘要」「复制永久链接」。
- **卡片②·交叉曲线**：一句话规律——模型越大，量化越可能省电。曲线在 0 以上 = 量化更费电；0 以下 = 省电。实线只连接实测点，超出实测范围的尺寸不在此图绘制。
- **卡片③·最新实测：RTX 4090（Ada）**：15/15 组配置全是真实硬件测量（0.5B–7B × FP16/NF4/INT8），由我们的开源容器在自己租的 RTX 4090 上跑出（2026 年 7 月），**每个配置 n = 1**。

这一列的关键发现（Δ 全部由能量列重算，见下方「更正说明」）：

| 模型 | NF4 vs FP16 | INT8 vs FP16 |
|---|---|---|
| Qwen2-0.5B | **+35.6%** | **+241.9%** |
| TinyLlama-1.1B | **+16.3%** | **+146.1%** |
| Qwen2-1.5B | **+15.1%** | **+180.7%** |
| Qwen2.5-3B | **+0.8%** | **+134.8%** |
| Qwen2-7B | **−28.5%** | **+49.5%** |

- **NF4 在这张卡上 3B 就已打平**（+0.8%），7B 是 −28.5% 的净节省；3B→7B 插值得交叉点约 **3.1B**。首页拟合曲线给的是 **≈3.7B**，因为拟合里还包含惩罚更重的 RTX 4090D 锚点。
- **INT8（bitsandbytes LLM.int8()）在我们测过的每一个尺寸上都没能省电**（+50%…+242%，五个尺寸、一张卡、每个 n = 1）：它确实降低瞬时功耗，但解码吞吐掉到 9–19 tok/s（FP16 为 39–62），每个 token 反而更贵。这是「**这个后端在这张卡上**」的结论，不是对 INT8 这个格式的判决——换 GPTQ/AWQ/TensorRT-LLM 或别的卡完全可能相反。
- 原始 `energy.json`、汇总 CSV、环境元数据都存档在 Zenodo（DOI 10.5281/zenodo.21528103，CC BY 4.0）。

**更正说明（2026-08-11）：**该 Zenodo 记录 v1 的 `vs_fp16_pct` 一列与它自己的能量／功率／吞吐列对不上（最严重的是 Qwen2.5-3B NF4：列里写 +10.1%，能量算出来是 +0.8%）。**测量值本身没有改变**，站点的 `measured.csv` 与所有曲线一直都是从能量列推导的；上表也已按能量列重算。修正版 Zenodo 记录待发布，详见 那篇 4090 笔记。

## 3 · 标签二：你的模型（Your model / 估算）

功能：输入你关心的模型规模，估算量化后的能耗变化与不确定性。

- 模型规模滑块（参数量，单位 B）：唯一必填项。
- FP16 区指示器（ecoGate）：标出「还没到交叉点、量化会赔」的区域。
- 高级选项（可展开）：GPU 架构、精度、批大小、上下文长度。
- **卡片①**：大数字 = 估算的能耗增减百分比，带置信区间（CI）与数据基础标签（实测锚定 / 拟合外推），同时给出估算 VRAM、估算能耗、结论与注意事项。
- **卡片②**：把你输入的模型标在拟合曲线上并画出 ±CI 误差带。**注意这是拟合外推，不是新测量**；大模型（≈≥7B）的结论只作方向性参考。
- 可「复制本估算的永久链接」，并展开看 JSON 报告结构。

## 4 · 标签三：优化（约束）（Optimize）

功能：从「静态查询」升级到「带约束的优化」——在预算内找最优配置。

- 三个输入：GPU 架构、模型规模、优化目标（最小化能耗 / 延迟、最大化吞吐、最小化显存）。
- 高级约束（可选）：最大延迟、最大显存、最小吞吐。
- 工具会穷举 精度 × 批大小 × 上下文，按约束过滤，返回推荐配置、备选方案，以及能耗 ↔ 延迟帕累托前沿（左下角更优，★ 为推荐点）。

## 5 · 标签四：自己跑（Run it yourself）

功能：在自己的 GPU 上复现测量，把结果叠加到站点的交叉曲线上。详见 容器页。

- **方法一·有 Docker（一行命令）**：直接 `docker run` 预构建镜像，无需下载执行脚本，命令完全透明。换精度用 `ECOCOMPUTE_PRECISION=INT8`；指定某张卡用 `--gpus "device=1"`。
- **方法二·无 Docker（AutoDL / vast.ai 等租卡）**：一行 `curl … | bash` 会自动克隆仓库、建 venv 并运行（租卡通常不能嵌套 Docker，这条路才需要）。**第一次跑要装 torch 等依赖，实测约 57 分钟**（缓存后重跑约 2 分钟）——我们不承诺 15 分钟。
- **方法三·从源码（完全可控）**：`git clone` + MLCube / `python3 entrypoint.py`。
- **你得到什么**：与本站字段完全一致的 `energy.json`；只有真实 NVML 采到功耗时 `basis` 才是 `measured`，采不到会明确标成回退值，不要拿去发布。
- **叠加结果**：把 `energy.json` 拖进上传区，你的点会出现在交叉曲线上——全程在浏览器本地解析，不上传任何数据。运行结束还会打印一个 `quantenergy.tech/?tab=run&overlay=…` 链接，打开即恢复你的标记。同意或不同意，都可以在 复现页 提交。

## 6 · About 弹窗

- **核心发现**：权重量化（NF4/INT8）减小显存占用，但不总是降低能耗；小模型上反量化开销会抵消带宽收益 → 更费电，大模型才开始省。交叉点取决于模型规模 + GPU 架构两者。
- **我们怎么测**：直接 NVML 功耗采样（10 Hz、256 token/run、10 次解码迭代、2 次预热），跨 4 种架构。**10 次迭代不等于 10 次独立试验**——一份报告仍然是 n = 1。建模部分（延迟/吞吐用 roofline 模型、超范围用外推）会明确标注，绝不把估算当测量。
- **我们自己的测量容器**：EcoCompute energy MLCube，开源、能跑在你的 GPU 上；最新的 RTX 4090 列就是它产出的。（使用 MLCube 规范不代表 MLCommons 背书，也不是官方 MLPerf 结果。）
- **研究论文**：SSRN #6854700（含 Zenodo 存档 DOI）。
- **该引用哪个产物？**（三个独立产物，各管一摊，完整对照见 引用页）
  - 论文 → SSRN #6854700
  - 主数据集 v1.1.0（360+ 配置，0.5B–14B，4 架构，n=2，CV<2%）→ DOI 10.5281/zenodo.19647290（本页除 RTX 4090 列外的数据都以此为准）
  - RTX 4090（Ada）深挖（2026 年 7 月，15/15 实测，n=1）→ DOI 10.5281/zenodo.21528103

- **给开发者**：REST API（`/v1/estimate`、`/v1/optimize`）、OpenAPI、供 Claude/Cursor 使用的 MCP 工具 `recommend_config`。该 API 是可选的便利接口，站点本身不依赖它。

## 7 · 页脚与博客

- 数据署名：Hongping Zhang；数值来自推理期间的直接 NVML 功耗采样（非 TDP 估算）；能耗按每 100 万 token 计，ΔE% 相对各自 FP16 基线。
- 链接：博客／笔记（/blog/）、RSS、GitHub、数据集与代码。
- 引用框：可展开 BibTeX，含论文、主数据集 v1.1.0、RTX 4090 实测三项 DOI。

## 8 · 手机上最该记住的 4 条「诚实声明」

1. **量化 ≠ 省电**：小模型量化往往更费电，大模型才省；交叉点随 GPU 架构变化。
2. **测量 / 估算分得清**：柱状图与「已测模型」是实测；「你的模型」是估算，带置信区间，大模型只作方向参考。
3. **知道每个数字的边界**：GPU 芯片功耗、单流 batch 1、特定后端版本；RTX 4090 那一列每个配置 n = 1，没有误差棒可画。同一配置隔一次会话重跑，我们实测到的差异可达 8.2 个百分点（1.1B INT8：+146.1% 与 +137.9%）。
4. **隐私**：上传 `energy.json` 只在你浏览器本地解析，不上传服务器；分享链接把结果编码在 URL 里。

---

本页是 quantenergy.tech 的中文说明，内容以站点当前数据为准；英文版说明散布在 方法页、容器页 与 引用页。
论文：SSRN #6854700 · 主数据集 v1.1.0：10.5281/zenodo.19647290 · RTX 4090 深挖：10.5281/zenodo.21528103 · 测量容器