

Heterogeneous Low-Cost Distributed Probing for LLM Observation

Chen Ho Yiing (Norika Oda)
Independent Researcher
ORCID: 0009-0006-6816-9891

Draft v0.1
2026-05-30

Abstract

I describe a distributed probing protocol that ran a 100M-cell dense activation grid on thirteen open LLMs (124M to 7.6B parameters) using a small cluster of personal machines: a main workstation plus a few older laptops and an occasional accelerator, with a remote archive node holding the merged output. No cloud burst. No data-center GPU. Calibration runs once on the host with enough RAM to hold the model; heat-fill runs in additive uint32 shards across whatever workers are available. The merge step uses element-wise sum and rejects shards with mismatched inputs by hashing. Wall-clock model: $T = T_{\text{calib}} + (T_{\text{fill}} \cdot P)/W$ for W workers on P probes. Heat tensor memory: $|H| = L \cdot d \cdot Q \cdot S \cdot 4$ bytes, where $L \cdot d \cdot Q \cdot S \geq 10^8$. On commodity CPU at $P = 1200$ and bfloat16, a 7B-class model completes in approximately one hour of calibration plus four worker-hours of heat-fill, all within ~ 400 MB of dense grid memory.

Keywords: distributed probing, heterogeneous compute, additive shard merge, dense activation grid, low-cost interpretability, consumer hardware, Mercury, independent research, reproducibility, open source

1 The setup

I work with a small set of personal machines: one main workstation that holds the model, a couple of older personal laptops that contribute compute in the background, and an occasional accelerator node when one is available. A remote archive node holds the merged output. All nodes coordinate over a private mesh VPN.

The question this paper answers: with this kind of setup, can I produce a coverage-qualified dense activation grid on a 7B-class model? The answer is yes, with the protocol below, in about one hour of calibration plus a few worker-hours of heat-fill.

This is an engineering report. The mechanistic findings produced on top of this substrate are reported elsewhere [8, 9, 10].

2 Related Work

Dense activation probing tooling has converged on hooks-based libraries that operate on a single accelerator [1, 2]. My prior work established the addressable cell grid concept [3], browser-resident

visualization [4], and a 23-architecture observation database [5]. The protocol described here scales the underlying observation pass across heterogeneous workers; the coverage criterion is from [8], and the per-channel quantile binning that the protocol applies on top is from [9].

Cross-layer fingerprint alignment across architectures was established in [6], and the methodology paper on Tier-A output-layer artifacts in [7]. Neither addressed the engineering question of how to actually run dense probes on commodity heterogeneous compute.

3 Notation and protocol

3.1 Grid

For a model with L transformer layers and d hidden width, with M hooked submodules total, the heat tensor is

$$H \in \mathbb{N}^{M \times d_{\max} \times Q \times S}, \quad H[m, c, q, s] \in \mathbb{N}, \quad (1)$$

In this notation, Q denotes the number of quantile buckets per channel, S the number of sampled sequence positions, $d_{\max}$ the maximum per-module output dimension, and the entry $H[m, c, q, s]$ counts how often the activation at module m , channel c , sequence-position bucket s has fallen into quantile bucket q across all probes.

Cell count target: $|H| \geq 10^8$. Memory at uint32:

$$\text{mem}(H) = 4 \cdot |H| = 4 \cdot M \cdot d_{\max} \cdot Q \cdot S \text{ bytes}. \quad (2)$$

For $|H| = 10^8$ this is approximately 400 MB; trivially fits in commodity RAM.

3.2 Two passes, one corpus

My pipeline runs two forward-pass sweeps over the same probe corpus of P prompts. Calibration first:

1. For each prompt $p \in \{1, \dots, P\}$, run a forward pass with hooks on all hooked submodules.
2. From each hook, sample activations at S evenly spaced sequence positions.
3. Accumulate per-channel activation samples up to a cap of R rows per module (memory bound).
4. After the sweep, compute $Q + 1$ quantile edges per (module, channel) pair: $\mathcal{E}_{m,c} \in \mathbb{R}^{Q+1}$.
5. Persist edges as a single npz archive: $\mathcal{E} = \{\mathcal{E}_{m,c}\}_{m,c}$.

Heat-fill second:

1. For each prompt p , run the forward pass again.
2. For each module m , channel c , sequence-position bucket s , digitize the activation against $\mathcal{E}_{m,c}$ to get a quantile bucket q , and increment $H[m, c, q, s]$.

3.3 Sharding

The fill pass is additive in P . Partition $\{1, \dots, P\}$ into W groups of approximately equal size, then assign one group per worker. Each worker w produces its own heat tensor $H^{(w)}$.

Merge:

$$H = \sum_{w=1}^W H^{(w)}. \quad (3)$$

Element-wise integer addition. A dedicated host loads each $H^{(w)}$ as memory-mapped uint32 and accumulates into a single output tensor. No re-computation, no reduction tree.

3.4 Invariants the merge enforces

Additivity requires all shards to share identical inputs. The merge step computes a SHA-256 hash over each of the following and refuses to merge mismatched shards: the safetensors blob containing model weights, the probe corpus file, a tuple combining $(Q, S, \text{include_norms})$, the computation dtype (`bf16`), and the quantile edges archive $\mathcal{E}$. A single byte difference in any of these breaks additivity, and the merge aborts with an error message naming the offending field.

4 Wall-clock model

Let $t_{\text{fwd}}(p)$ be the wall-time for a single forward pass on probe p . Calibration runs sequentially on one host:

$$T_{\text{calib}} = \sum_{p=1}^{P_{\text{cal}}} t_{\text{fwd}}(p) + t_{\text{edges}}, \quad (4)$$

Here $P_{\text{cal}} \leq P$ denotes the calibration corpus size (typically $P_{\text{cal}} = 256$), and t_{edges} is the per-channel quantile computation cost, $O(M \cdot d \cdot R \log R)$, which is small in practice.

The fill pass runs in parallel across W workers:

$$T_{\text{fill}} = \frac{1}{W} \sum_{p=1}^P t_{\text{fwd}}(p) + t_{\text{digitize}}(p), \quad (5)$$

Here t_{digitize} denotes the per-channel digitization overhead, approximately 10% of forward pass time for per-channel binning. Element-wise accumulation in the final aggregation step adds a small term:

$$T_{\text{merge}} = (W - 1) \cdot |H| \cdot t_{\text{add}}, \quad (6)$$

Here t_{add} denotes the per-cell addition time (sub-nanosecond on modern memory). For $|H| = 10^8$ and $W = 4$, this is seconds.

Total wall-clock:

$$T = T_{\text{calib}} + T_{\text{fill}} + T_{\text{merge}} \approx T_{\text{calib}} + \frac{1}{W} \cdot 1.1 \cdot \sum_{p=1}^P t_{\text{fwd}}(p). \quad (7)$$

Heat-fill parallelises, but calibration does not. The asymmetry is intentional: calibration must produce a single coherent $\mathcal{E}$ that all workers share, and parallelising it would require a reduction over per-channel quantile samples that defeats the additive merge guarantee.

5 Resource sufficient condition

A worker w is sufficient for participation if and only if its RAM exceeds the model’s bfloat16 footprint plus the per-shard heat tensor:

$$\text{RAM}(w) \geq 2 \cdot N_{\text{params}} + 4 \cdot |H| + C_{\text{overhead}}, \quad (8)$$

Here N_{params} is the parameter count (each bfloat16 weight is 2 bytes), the $4 \cdot |H|$ term is the per-shard uint32 heat tensor, and C_{overhead} covers ML-runtime overhead (typically 2–4 GB).

Concrete envelopes by model size:

- BLOOM-560M requires roughly $2 \cdot 1.1 + 0.4 + 3 \approx 5.6$ GB, so any modest personal laptop participates comfortably.
- Qwen2.5-7B requires roughly $2 \cdot 15.2 + 0.4 + 3 \approx 34$ GB. A mid-range laptop with ≥ 24 GB participates in bfloat16; below that the worker does not qualify. The main workstation and any accelerator with ≥ 64 GB fit comfortably.
- A 70B-class model requires roughly $2 \cdot 140 + 0.4 + 3 \approx 145$ GB. At this scale only a workstation-class node with ≥ 150 GB or a unified-memory accelerator of comparable capacity participates, and either path requires careful memory accounting if the budget is shared between CPU and GPU regions.

Worker dropout degrades gracefully: if a node fails the inequality in Eq. (8), it does not participate, and the remaining workers absorb its probe partition. Heat-fill time increases inversely with W , but the run still completes.

6 Heterogeneous worker assignment

Workers vary in t_{fwd} by a wide margin. A consumer laptop running an alternative ML runtime can be 2–4 $\times$ slower than a workstation CPU on the same model in bfloat16, depending on architecture and runtime op coverage. An accelerator node, when available, is the fastest by far during its engaged window.

A simple allocation strategy: estimate $t_{\text{fwd}}^{(w)}$ per worker on a 10-probe pilot, and allocate probe partitions proportional to inverse forward time:

$$\text{share}(w) = \frac{1/t_{\text{fwd}}^{(w)}}{\sum_{w'} 1/t_{\text{fwd}}^{(w')}}, \quad |P_w| = \text{round}(P \cdot \text{share}(w)). \quad (9)$$

This minimizes makespan under uniform-cost forward passes. In practice, I assigned the slowest worker the smallest probe partition, and the fastest the largest, and accepted whatever imbalance remained.

7 Operational observations

7.1 Bfloat16 throughout

Model weights, activations, and forward-pass arithmetic all in bfloat16. This keeps the model footprint at $2N_{\text{params}}$ bytes and matches the dtype released by upstream model publishers, which avoids spurious quantization artifacts in calibration.

7.2 Per-channel edges archive size

For a 7B model with $M \approx 200$ hooked submodules and $d_{\max} \approx 4096$, the per-channel edges archive at $Q = 10$ is

$$4 \cdot M \cdot d_{\max} \cdot (Q + 1) \approx 36 \text{ MB at fp32.} \quad (10)$$

Trivial transfer cost between workers over the mesh VPN. The edges archive is calibrated once on the host and copied to each worker before heat-fill.

7.3 Module discovery

The probe runner registers forward hooks on every `nn.Linear` submodule and (where present) HuggingFace `Conv1D` submodules. GPT-2 uses `Conv1D` for its attention and MLP projections instead of `nn.Linear`; missing this in early prototypes silently dropped M from ~ 74 to a tiny number, which then collapsed coverage. Module discovery is now hash-validated against a known reference M per model architecture before calibration starts.

7.4 Merge integrity

Validation runs on the host that performs the merge, against the five inputs listed in Section 3.4. In practice the most common abort cause was a non-identical probe corpus: a worker had received a slightly older version of the prompts file. Recovery is to re-run that worker’s shard with the canonical file. No actual data corruption; the abort is the script doing its job.

8 Cost outcomes

For thirteen runs across BLOOM, BLOOMZ, GPT-2, Gemma-2, Phi-2, Pythia, Qwen2.5, and InternLM-2 at 124M to 7.6B parameters, the protocol produced coverage-qualified (`coverage_pct` ≥ 90) dense activation grids in approximately:

- Calibration: 5 minutes (small models) to 60 minutes (7B model on CPU).
- Heat-fill, single worker: 30 minutes (small) to 4 hours (7B on CPU).
- Heat-fill, four workers: 8 minutes (small) to 1 hour (7B).
- Merge: under 30 seconds for any model.

No model required GPU to complete. The occasional accelerator reduced wall-clock by approximately $5\times$ on the runs that used it, but was not necessary for completion. The practical floor of this approach is set by RAM, not by compute: the smallest worker satisfying Eq. (8) participated, and the others waited.

9 Discussion

9.1 What this protocol does not give

My pipeline produces a coverage-qualified observation substrate. It does not produce mechanistic interventions, model edits, or training updates. Downstream tools [8, 9, 10] apply analysis on top of the grid.

9.2 What this protocol does give

A reproducible path from a published model checkpoint to a dense activation grid, using compute most independent researchers can assemble: a main workstation plus a couple of older personal laptops. No cloud burst. No data-center accelerator. The cost floor is the time of the slowest worker.

9.3 Trade-offs accepted

Calibration is not parallelised. Adding workers does not reduce T_{calib} . For 7B-class models calibration can dominate total time when the host is CPU-only. The right place to add GPU is calibration, not heat-fill.

There is no streaming. Each worker holds the model in RAM throughout its shard. A streaming variant would let smaller workers handle larger models at the cost of significantly higher disk I/O; this was not pursued.

Mid-shard checkpointing is also absent. A worker that crashes loses its partial heat tensor and re-runs its full shard. For $P/W \approx 300$ probes per worker, the loss is bounded.

10 Limitations

1. Tested up to 7.6B parameters only. Extension to 12B and 22B classes should fit within the existing RAM envelopes, but I have not verified.
2. No causal verification of the observation substrate’s downstream utility is reported here. That belongs to the analyses built on top [8, 9, 10].
3. The wall-clock estimates assume per-channel quantile binning. Pooled binning is roughly 10% faster but produces lower coverage [9].
4. The hash validation in Section 3.4 is necessary but not sufficient: it catches accidental mismatch, not adversarial tampering. Reproducibility, not security.
5. Per-worker throughput numbers reflect the specific personal machines I had access to and the ML runtimes installed on them. Other consumer or accelerator hardware may differ substantially and is not benchmarked here.

11 Reproducibility

Implementation has two parts: a probe runner that accepts model path, probe corpus path, Q , S , dtype , and `include_norms` flag; and an additive merge script that takes a list of shard directories and produces the merged heat tensor. Both follow the calibration-then-fill structure described in Section 3.

A run completes when its `meta.json` reports `actual_cells` $\geq 10^8$ and `coverage_pct` ≥ 90 . The completion criterion is described in detail in [8].

Exact tool paths and the probes corpus file are available on request.

12 Follow-on

1. Scale-out: verify the protocol on Mistral-Nemo-12B and Codestral-22B class models.

2. Streaming variant: investigate whether memory-mapped model weights let smaller-RAM laptops participate in 7B-class runs at the cost of higher I/O.
3. GPU calibration: add an optional GPU-accelerated calibration pass to shrink T_{calib} for the cases where it dominates.
4. Cross-worker reproducibility: characterise the floating-point divergence between different CPU and accelerator backends on identical inputs, and add a tolerance band to the merge invariant check.
5. Probe corpus design: study how the choice of P probes affects coverage on different model families, with the protocol held fixed.

13 Acknowledgments

The compute substrate was assembled from one main personal workstation, a couple of older personal laptops, and an occasional accelerator node available to me during this study. A remote archive node serves as merge host and long-term storage.

References

- [1] Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Tamkin, A., Nguyen, K., McLean, B., Burke, J., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. Transformer Circuits Thread, Anthropic, October 2023. <https://transformer-circuits.pub/2023/monosemantic-features>.
- [2] Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy Models of Superposition. Transformer Circuits Thread, Anthropic, September 2022. https://transformer-circuits.pub/2022/toy_model/.
- [3] Chen, Ho Yiing. Mercury: An addressable million-cell sensor grid for browser-accessible LLM observability on consumer hardware. Zenodo, 2026. DOI: 10.5281/zenodo.20313154.
- [4] Chen, Ho Yiing. Mercury-Viewer: A self-contained browser-based interactive 3D visualization of LLM internal sensor maps. Zenodo, 2026. DOI: 10.5281/zenodo.20313150.
- [5] Chen, Ho Yiing. Mercury MCP v0.1.0 — Cross-Architecture LLM Internal Observation Database for 23 LLMs. Zenodo, 2026. DOI: 10.5281/zenodo.20352085.
- [6] Chen, Ho Yiing. Cross-Architecture Per-Layer Residual Stream Alignment via Tier-B Fingerprints: 84 Model-Pair Matrices Across 9 Open LLMs Show a Universal Middle-Depth Slot. Zenodo, 2026. DOI: 10.5281/zenodo.20366010.
- [7] Chen, Ho Yiing. Tier-A Vocab-Hash Artifact in Cross-Architecture Hot-Dim Studies: Why Output-Layer Token-ID Projection Manufactures False Universality, and an Artifact-Aware Screening Protocol. Zenodo, 2026. DOI: 10.5281/zenodo.20366134.
- [8] Chen, Ho Yiing. Coverage Failure as a Diagnostic Signal in Neural Model Probing. Zenodo, 2026. DOI: 10.5281/zenodo.20448669.
- [9] Chen, Ho Yiing. Per-Channel Quantile Fullgrid Probing for Small Language Model Observation. Zenodo, 2026. DOI: 10.5281/zenodo.20448674.

- [10] Chen, Ho Yiing. Relation-State Continuity in Qwen2.5-7B Base: Evidence for an Attention-Topology Path. Zenodo, 2026. DOI: 10.5281/zenodo.20449372.