

Mercury-Viewer: A Self-Contained Browser-Based Interactive 3D Visualization of LLM Internal Sensor Maps

Chen, Ho Yiing¹

¹Independent researcher, charenix.com, ORCID: 0009-0006-6816-9891
abbychen981008@gmail.com

2026-05-20

Abstract

This paper releases **Mercury-Viewer**, a single HTML file (2,031,982 bytes) that opens in any browser and shows the inside of a transformer language model as a 3D point cloud. No install, no server, no GPU, no API key. Mercury-Viewer reads activation data from the Mercury sensor grid [1] and draws it in cylindrical coordinates: azimuth is the hidden dimension index, radius is the activation quantile, height is how many prompts activated a given unit. Every point keeps a full (L, D, Q) coordinate, so hovering reveals the exact location, and the picture is invertible. Alongside the viewer, I release the **Mercury-3B-7B observation dataset**: 21,694 fired-cell records and 28,654 visible Rosen-bridge co-firing edges, captured by running ten multilingual prompts through qwen2.5:3B and qwen2.5:7B on a single RTX 3060. Code MIT, data CC-BY-4.0, both archived on Zenodo with a DOI. The whole pipeline ran in 24 hours on consumer hardware.

1 Introduction

Most LLM interpretability work happens in places I cannot get into. The dashboards sit behind institutional logins. The published analyses need GPUs I do not own. The notebooks ship with datasets too large to redistribute, citing infrastructure I have no path to.

So I built one for myself, and now I am giving it away.

The Mercury project [1, 2] produces per-prompt activation observations across an addressable million-cell sensor grid placed over a model’s hidden state space. One observation pass outputs under 50 MB of small binary files. Looking at that data should not require installing PyTorch, signing up for a service, or owning an A100.

Mercury-Viewer is the rendering half of that pitch. It reads the observation outputs and draws a 3D cylindrical scatter plot inside a single self-contained HTML file. Chrome, Safari, Firefox, Edge: all of them open it the same way. The file works offline. There is a lane filter and a hover inspector built in, so the viewer is both a published figure and a working tool at the same time.

This paper makes four contributions:

1. A self-contained, dependency-free, browser-based 3D viewer for LLM internal sensor maps.
2. A cylindrical coordinate system in which each rendered point uniquely identifies a (layer, dim, quantile) triple, so the visualization is invertible.
3. An open dataset of 21,694 fired-cell observations and 28,654 visible co-firing edges from qwen2.5:3B and qwen2.5:7B, enough to reproduce every figure in the Mercury method and discovery papers.
4. Permanent archival under MIT (code) and CC-BY-4.0 (data) with a Zenodo DOI.

2 Related Work

Existing visualization tools split into three groups. The first is architecture diagrams: tools like Bycroft’s LLM Visualization [7] and BertViz [6] draw the model graph, but they show no live activation. The second is activation atlases [8], which render learned features beautifully but require a trained dimensionality reducer (typically a sparse autoencoder [4]) as a preprocessing step that costs real GPU time. The third is library-level inspectors like TransformerLens [5], which expose hooks in Python; what you do with them is your problem.

Mercury-Viewer is closest in spirit to the atlas family, but differs on two axes. First, the cells are placed *a priori* by a grid scheme, not learned, so the coordinate system is fixed and the picture lines up across observations of different prompts. Second, the artifact is a single HTML file. Not a notebook. Not a web app. Not a repository with environment files. One file.

3 Architecture

3.1 Input data

A Mercury observation pass produces, for each of N prompts ($N = 10$ in the released dataset):

- `fired-cells-{tag}_{ts}_p{i}.bin`: a uint32 array of cell IDs whose activation exceeded the per-cell threshold during prompt i .
- `bridges-{tag}.bin`: the a-priori Rosen-bridge graph, packed as uint32 neighbor lists.
- `catalog-{tag}.bin`: a 24-byte-per-cell catalog giving each cell’s (L, D, Q) coordinate and quantile threshold.
- `manifest-{tag}.json`: grid metadata (layer count, hidden dim, quantile array, total cell count).

For qwen2.5:7B the catalog is 24,084,480 bytes describing 1,003,520 cells across $28 \text{ layers} \times 3584 \text{ dims} \times 10 \text{ quantiles}$. The per-prompt fired-cells files are tiny: 12 to 15 KB each, encoding 3000 to 3700 fired cells.

3.2 Coordinate transform

For each grid point that was active in at least one prompt, I compute three quantities:

$$\theta = 2\pi \cdot \frac{D}{H}, \quad r = Q + 1, \quad y = c,$$

where $D \in [0, H)$ is the hidden dimension index, H is the model’s hidden size, $Q \in [0, 10)$ is the quantile bucket, and $c \in [1, N]$ counts the prompts in which the unit was active. The point then lands at

$$(x, y, z) = (r \cos \theta, y, r \sin \theta).$$

Three properties fall out of this. The mapping $\theta \mapsto D$ is bijective, so a hovered point reveals an exact dimension index. The radius encodes activation strength, so cells near the outer surface of the cylinder are the most-extreme firings. And the height encodes the *breadth* of activation across prompts, so the so-called “universal backbone” cells form a visible ring at the top of the cylinder and topic-pure cells settle on the base disc. The picture sorts itself.

3.3 Rendering

The renderer ships Three.js [9] as inline JavaScript (about 600 KB after minification) and uses an instanced Points object for the point cloud. Edges from the Rosen-bridge graph are filtered before rendering: I keep only those whose *both* endpoints lit up in at least one prompt, which collapses 5,947,760 a-priori bridges down to 18,113 visible ones for the 7B grid. The visible edges go into one LineSegments buffer. Hover inspection runs through a GPU picking pass. Lane filtering is a per-vertex attribute toggle, so changing the filter does not rebuild geometry.

3.4 Self-containment

A small Python script reads the observation outputs, applies the coordinate transform, base64-encodes the resulting JSON, inlines it into an HTML template alongside the minified Three.js and the UI JavaScript, and writes a single .html file. The released file is 2,031,982 bytes for the 7B observation, including all 14,912 points, 18,113 visible bridges, lane assignments, and the rendering library.

When you open the file, no network requests go out. That is the test.

4 Dataset

The released dataset, **Mercury-3B-7B-Observation-v1**, comprises the artifacts in Table 1. Total size on disk is 49.1 MB.

Table 1: Released dataset contents.

File	Contents	Size
mercury-3d-viewer.html	Self-contained 3D viewer (7B observation)	2.03 MB
catalog-1m.bin	1,003,520-cell catalog for qwen2.5:7B	24.08 MB
bridges-1m.bin	A-priori Rosen-bridge graph (7B)	23.79 MB
fired-cells-1m*_p{01..10}.bin	Per-prompt fired-cell IDs (7B), 10 files	138 KB
catalog-3b.bin	368,640-cell catalog for qwen2.5:3B	8.85 MB
bridges-3b.bin	A-priori Rosen-bridge graph (3B)	8.74 MB
fired-cells-3b*_p{01..10}.bin	Per-prompt fired-cell IDs (3B), 10 files	78 KB
manifest-{1m,3b}.json	Grid metadata	1 KB
observation-*.jsonl	Per-prompt wall-time and summary	4 KB
heat-summary-*.json	Top-100 hottest cells per grid	12 KB
visible-bridges.json	Pre-filtered visible edges (both endpoints fired)	519 KB
README.md, LICENSE	Reproduction recipe + licensing	8 KB

The ten prompts span four topic groups (physics, algorithms, machine learning, casual) crossed with two languages (Chinese, English), matched one-for-one between the 3B and 7B runs. The prompt list ships in full as `prompts.json`. I picked the smallest prompt set that still resolves a language axis and a topic axis under signature clustering [2]; ten prompts is what I landed on.

4.1 Observation statistics

Table 2 gives the headline numbers. Both observations were collected in a single sitting on an RTX 3060 12 GB on 2026-05-20, commits `fd43cd8` (7B grid build) and `8ee6256` (3B build).

Table 2: Mercury-3B-7B observation summary.

Quantity	qwen2.5:3B	qwen2.5:7B
Total grid cells	368,640	1,003,520
Layers $\times$ dim $\times$ quantile	$36 \times 1024 \times 10$	$28 \times 3584 \times 10$
Cells fired (any prompt)	6,782	14,912
Distinct firing signatures	612	944
A-priori bridges	2,211,840	5,947,760
Visible bridges	10,541	18,113
Observation wall time	2.4 min	3.5 min
Build wall time	0.9 s	2.49 s

One number in that table is worth flagging. The 7B build renders 14,912 active points and 18,113 visible bridges on integrated GPUs at interactive frame rates. That leaves plenty of headroom for Tier-B, where a 28-layer pass produces an order of magnitude more activated units. The bottleneck is not rendering. It never was.

5 Use Cases

5.1 As a published figure

Linking to the hosted viewer at <https://charenix.com/static/neuron-mercury/llm-model.html> from a paper or blog post lets a reader manipulate the visualization themselves: rotate the cylinder, toggle the universal-lane filter to see the 39-cell backbone in isolation, hover over the topmost ring to read off the exact dimension indices that fired in all ten prompts. A static screenshot loses all of that. A live viewer does not.

5.2 As a hand-off artifact for collaborators

Mail the 2 MB .html file. That is the whole hand-off.

The recipient does not need access to the model, the grid catalog, or the raw fired-cell binaries to reproduce the picture. For collaborators who want to run their own queries, the raw binaries in the dataset record are enough.

5.3 As a teaching tool

The cylindrical layout makes the layer-27 output ring visible at first glance: it dominates the figure because the placement scheme allocates ten quantile buckets to each (L, D) pair. A natural classroom exercise is to ask students what changes when the model goes from 3B to 7B, then let them check their prediction against the side-by-side viewer. The answer is more interesting than people expect.

6 Reproducibility

The viewer and dataset together suffice to reproduce every numeric claim in this paper. End-to-end on one RTX 3060 takes under ten minutes for the 7B observation:

```
git clone https://github.com/norika/Neuron-Mercury
cd Neuron-Mercury
ollama pull qwen2.5:7b
python3 scripts/build-grid-1m.py
python3 scripts/observe-grid-1m.py \
    --model qwen2.5:7b --max-tokens 100
python3 scripts/build-3d-viewer.py
# open the produced HTML file in any browser
```

The same pipeline with qwen2.5:3b reproduces the 3B half of the dataset. The Zenodo record bundles a frozen copy of all scripts at the commit hashes cited above. Against those frozen scripts, the catalog and bridge files are byte-for-byte deterministic. The fired-cell files depend on the model’s quantized weights and on sampling temperature: the temperature is fixed, but bit-exact reproduction across different hardware is not something I claim.

7 Limitations

The released observation is single-output-layer, what I have been calling “Tier-A” in the Mercury notes. Only the residual stream at the final transformer block is observed. Earlier layers exist in the grid catalog but went unobserved in this release, because the Python observer hooks into `logits_processor`, which sees only the last layer’s output. A Tier-B build, requiring a patched `llama.cpp`, is in preparation and will ship as a separate dataset record. The viewer architecture already supports Tier-B data without changes; the bottleneck is the observer, not the renderer.

The observation also covers one model family (Qwen2.5) at two sizes. Multi-family comparison (Llama-3, Mistral, Yi) is out of scope for this release.

I am aware of both gaps. Both are on the roadmap.

8 Availability

- **Live viewer:** <https://charenix.com/static/neuron-mercury/llm-model.html>
- **Source code:** <https://github.com/norika/Neuron-Mercury> (MIT)
- **Dataset record:** Zenodo DOI to be minted upon submission; CC-BY-4.0
- **Companion papers:** method [1], discovery [2], scaling [3]

9 Acknowledgements

The Mercury observation pipeline, the Rosen-bridge topology, and the drafting of this paper were completed in a 24-hour sprint across 2026-05-19 and 2026-05-20, in collaboration with an Anthropic Claude Code session acting as a pair-engineer. Every numeric claim in this paper traces back to a git commit hash and an on-disk binary. The full provenance map lives in `data-integrity.md` inside the Zenodo record. If a number in this paper does not trace, that is a bug, and I want to know.

References

- [1] Chen, Ho Yiing. Mercury: An addressable million-cell sensor grid for browser-accessible LLM observability on consumer hardware. Zenodo, 2026. DOI: 10.5281/zenodo.20312638.
- [2] Chen, Ho Yiing. Unsupervised discovery of language and topic lanes in transformer models via multilingual co-firing signatures. In preparation, 2026.
- [3] Chen, Ho Yiing. Internal specialization scales with model size: Quantitative evidence from qwen2.5 3B vs 7B vs 14B. In preparation, 2026.
- [4] 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>.
- [5] Nanda, N. and Bloom, J. TransformerLens. <https://github.com/TransformerLensOrg/TransformerLens>, 2022.
- [6] Vig, J. A Multiscale Visualization of Attention in the Transformer Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy, 2019. DOI: 10.18653/v1/P19-3007.
- [7] Bycroft, B. LLM Visualization. <https://bbycroft.net/llm>, December 2023.
- [8] Carter, S., Armstrong, Z., Schubert, L., Johnson, I., and Olah, C. Activation Atlas. *Distill*, 4(3):e15, 2019. DOI: 10.23915/distill.00015.
- [9] Cabello, R. (mr.doob) et al. Three.js: JavaScript 3D library. <https://threejs.org>, 2010–present.