operandus

Tribe setup guide.

Run your own private reel-analysis board with TRIBE v2.

A free, step by step guide for a developer or marketer.

Free and non-commercial. This guide is free. TRIBE v2 is by Meta FAIR and is licensed CC BY-NC 4.0, so what you build with it must stay non-commercial. Built with Llama. Not affiliated with or endorsed by Meta. See section 11.

MADE BY OPERANDUS

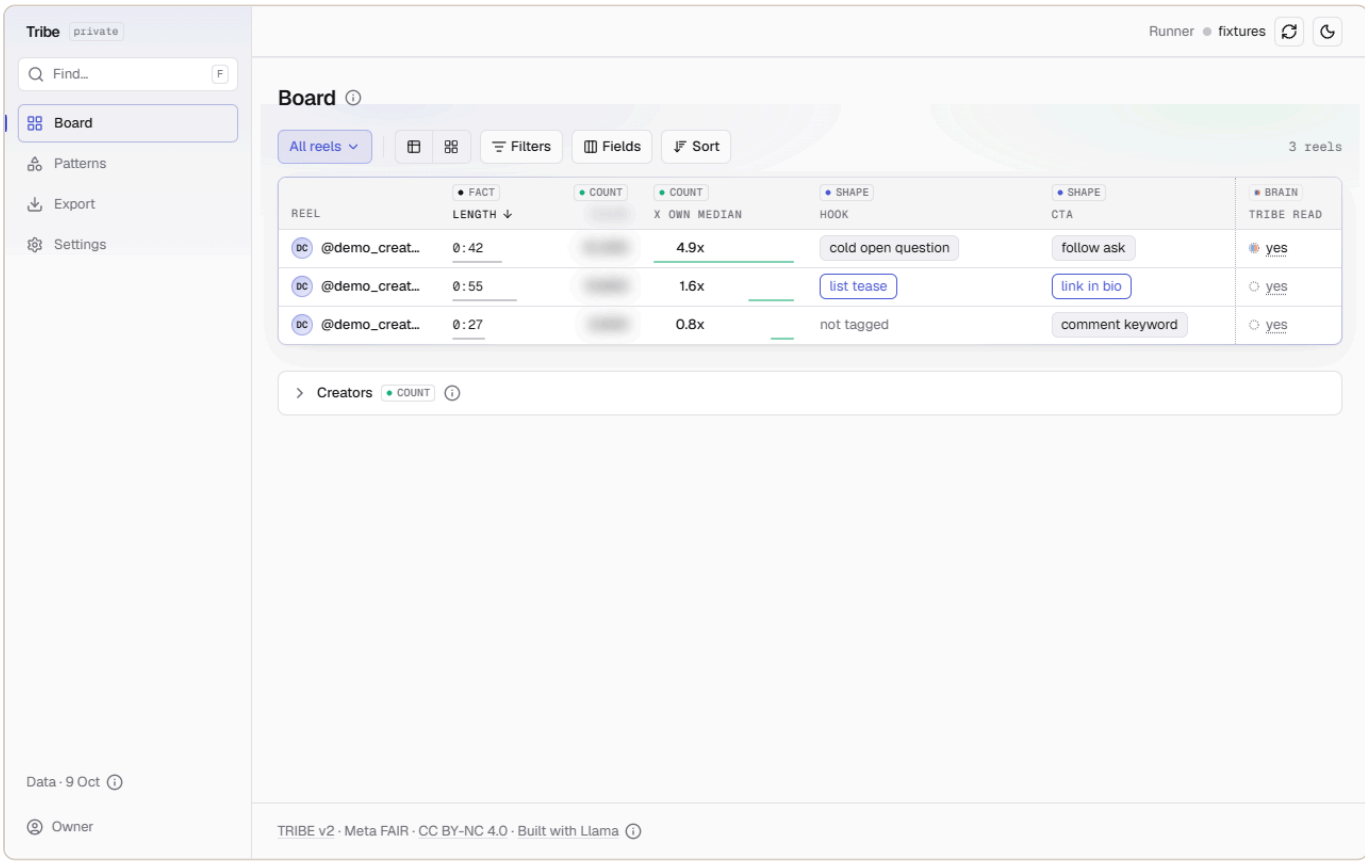
All screenshots show made-up demo data (fixtures), not real creators.

1 What you get

A private web board for a swipe file of short reels, running on accounts you own.

- **Board:** each reel next to its own creator's usual reel, with shape tags (hook, format, call to action).
- **Patterns:** counts of shapes among the outlier reels.
- **Reel page:** the video, a timeline, and for reels you choose, a 3D brain view of the **predicted average-brain response** from TRIBE v2.
- **Areas tab and CSV export:** seven standard brain networks and 180 cortical areas, in a fixed order.
- **Jobs page:** shows spend and limits, and lets you pause paid work.

It is analysis only. It does not write scripts or content, and it does not forecast views or performance.



Board (demo data). Each row is a reel; "TRIBE read" says whether a brain read exists.

2 What it costs

These figures come from the original build's own runs. Treat them as estimates and check each provider's current pricing page.

ITEM	COST	NOTE
Brain read (Modal GPU)	about $0.10 per read	One 10 to 60 s clip. Modal bills per second of GPU. The GPU runs the model, then exits.
Add a creator (Apify)	about $0.08 per creator	Fetches the creator's recent reels.
Add one reel by link (Apify)	about $0.003	Public reels only.
Vercel, Neon, Upstash	free tiers	Enough for a personal board. Vercel Blob (private video storage) used about 290 MB of its 1 GB in the original build.
Hugging Face	free	Needed only to download the model weights once.

Set a monthly cap before your first run. Idle GPUs are the real money risk, not single reads. The original build used a $30 monthly cap, logged every run, and kept no GPU warm.

3 Accounts to create

SERVICE	USED FOR	SIGN UP
Modal	Runs the TRIBE v2 model on a GPU	https://modal.com/signup
Neon	Postgres database	https://neon.com
Upstash	Redis for job wake-up and runner heartbeat	https://upstash.com
Vercel	Hosts the dashboard (with login)	https://vercel.com/signup
Apify	Fetches public reel data and videos	https://apify.com
Hugging Face	Model downloads	https://huggingface.co/join

Gated model access (do this first, it can take time)

1. Log in to Hugging Face, open **https://huggingface.co/meta-llama/Llama-3.2-3B** and request access. Use only this official page. Never use a re-uploaded copy of the model.
2. Open **https://huggingface.co/facebook/tribev2** and read its model card and licence.
3. Create a read-only access token under your Hugging Face account settings, Access Tokens.

4 Get the code and set up your keys

You need Git, Node.js 20 or newer, Python 3.11 or newer and the Modal CLI (`pip install modal`).

```
git clone https://github.com/operanduswork-cmd/tribe-synthesizer tribe
cd tribe
cp .env.example .env
```

Open `.env` in an editor and fill in the placeholders. The names below are examples. Use the names in the repo's `.env.example`.

```
HF_TOKEN=<your Hugging Face read token>
APIFY_TOKEN=<your Apify API token>
MONTHLY_BUDGET_CAP_USD=<your monthly GPU cap, e.g. 30>
TRIBE_WEB_DATABASE_URL=<Neon read-only connection string>
TRIBE_OWNER_DATABASE_URL=<Neon owner connection string, local only>
UPSTASH_REDIS_REST_URL=<from the Upstash console>
UPSTASH_REDIS_REST_TOKEN=<from the Upstash console>
```

Never commit `.env`, paste it into a chat, or screenshot it. Check that `.gitignore` lists it before your first commit.

5 Deploy the model on Modal

1. Log in with `modal setup`. Check you are on your own workspace with `modal profile current`.
2. Store the keys as Modal secrets, not in code: the Hugging Face token (for the weight download only) and your monthly cap value.
3. Run the account check, then the cheap CPU smoke test, then the one-time weight download. These are the `check`, `smoke` and `weights` steps of the repo's backend script (`backend/tribe.ps1` on Windows, or the equivalent `modal run` commands elsewhere).
4. The weights land in a Modal volume. You only download them once.

The GPU function runs inference and exits. Leave keep-warm off. Start with the cheapest GPU that fits (A10G in the original build) and move up only if it fails.

6 Run the dashboard

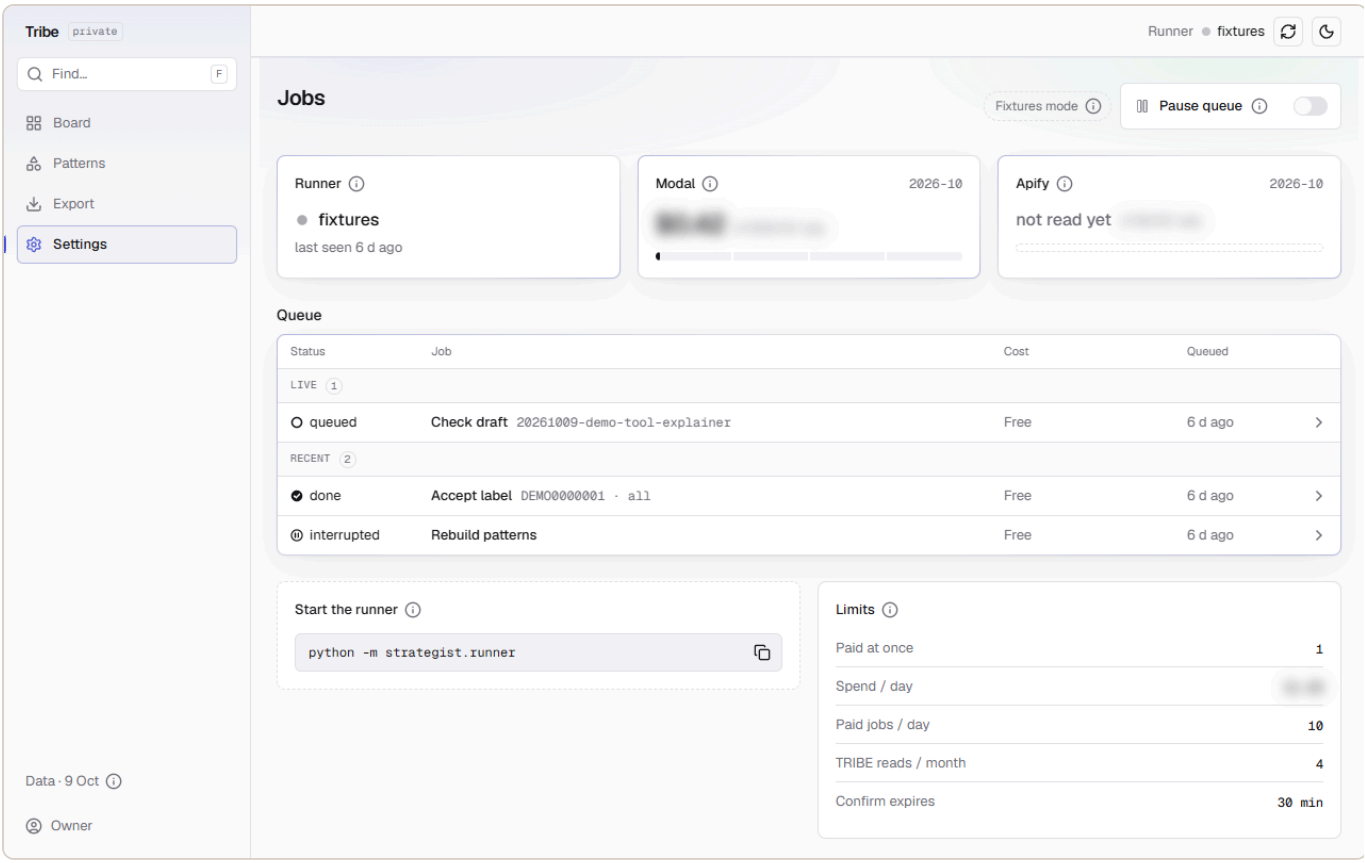
Try it with demo data first (free)

```
cd web
npm install
npm run build
TRIBE_DATA_SOURCE=fixtures npm run start -- -p 3100
```

Open `http://127.0.0.1:3100/board`. You will see made-up creators and reels, as in this guide's screenshots. Nothing is read from a database.

Then connect your own data

1. Create a Neon project. Run the repo's database migration with the owner connection string (`npm run db:migrate`). Use the repo's role script to make a read-only role for the website.
2. Create an Upstash Redis database.
3. Create a private Vercel Blob store for videos.
4. Deploy to Vercel with your own account, set `TRIBE_DATA_SOURCE=db` and the read-only connection string as environment variables, and **turn on Vercel Authentication** so every page asks for a login.
5. Start the local runner that does the paid work from your computer: `python -m strategist.runner`.



Settings, Jobs page: runner status, Modal and Apify spend against your caps, the queue and limits. Set the limits low.

7 Add a creator

```
python -m strategist.reel add @handle --yes
```

This costs about $0.08 and pulls the creator's recent public reels onto the board. To add one reel, pass its link instead (about $0.003). You can also paste a link or handle on the Import page, which shows the cost in a confirm box first. Use public accounts only.

8 Run a brain read

1. Pick a reel that is **10 to 60 seconds long and has speech** (see section 9).
2. Fetch its video to your computer with the repo's media command (`python -m strategist.media fetch`).
3. Send it to Modal: `backend/tribe.ps1 reel media/<shortcode>.mp4 -Label my-first-read`. This is the step that costs about $0.10.
4. Run the scoring script on the output folder, then link the result to the reel with `python -m strategist.reel link-tribe`. Follow the repo's README for the exact arguments.
5. Sync to the site and reload the reel page. "TRIBE read" on the board now says yes.

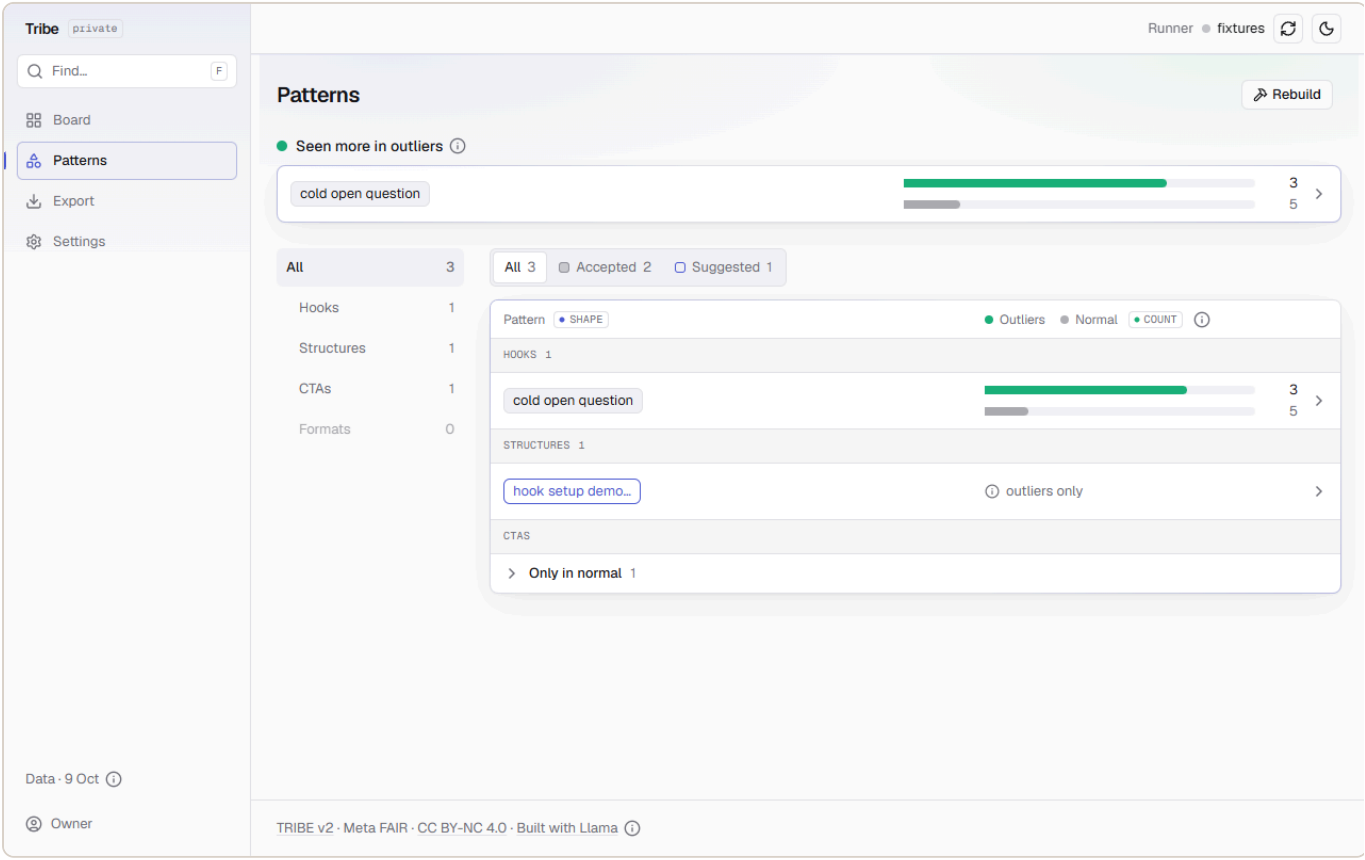
9 Clip rules

RULE	WHY
10 to 60 seconds	Under about 10 s the response comes out flat. Over 60 s the audio path repeats words (a known upstream bug).
It needs speech	Silent or music-only clips leave out the language path.
Real video, not a still image	Images must be turned into short clips and are far from what the model was trained on.
Only videos you may use	Keep copies private. Do not republish them.

10 How to read the board and the brain view

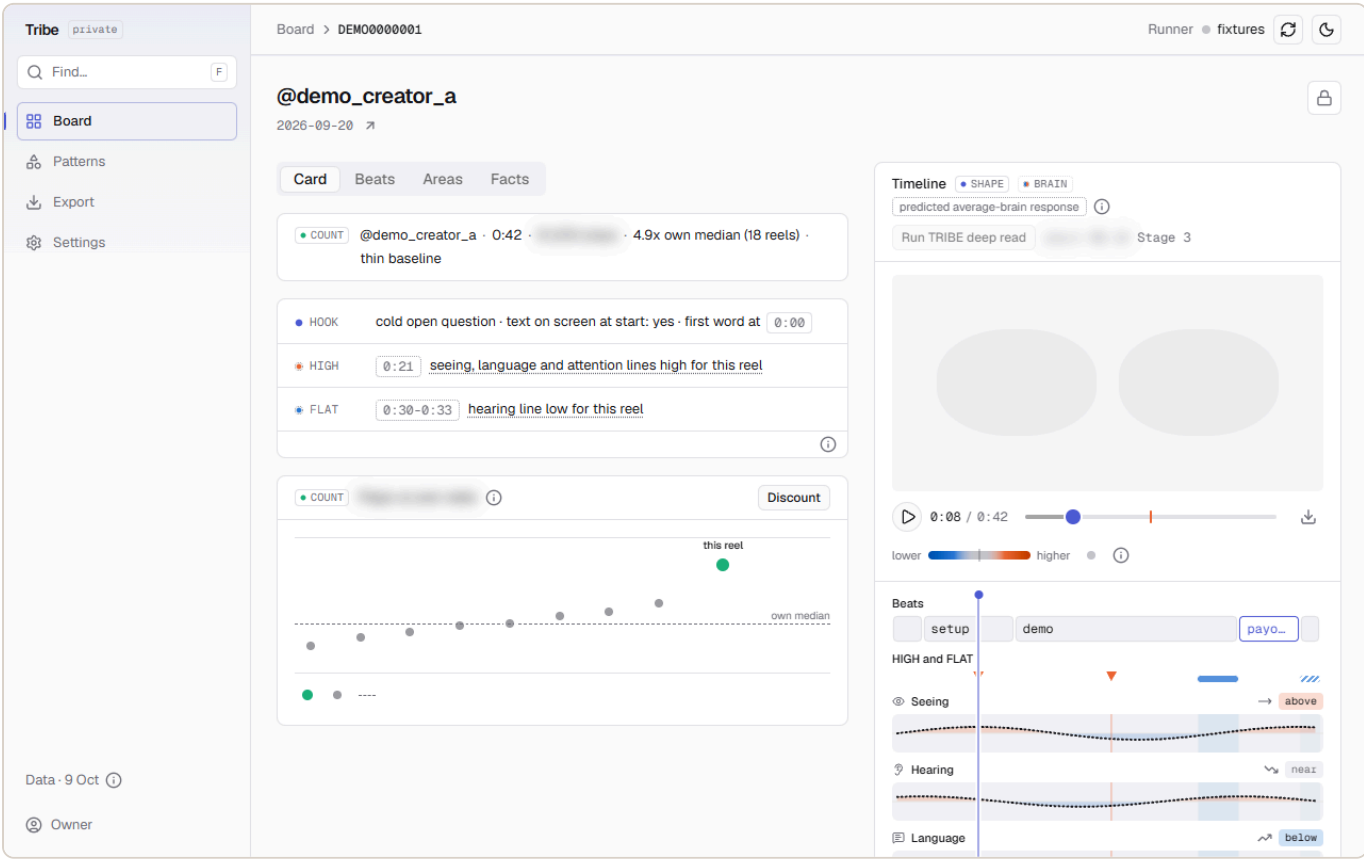
Board and patterns

"x own median" compares a reel with its own creator's usual reel. It is a count, not a quality grade. Patterns count shapes among outlier reels only, so treat them as ideas to test, not proof.



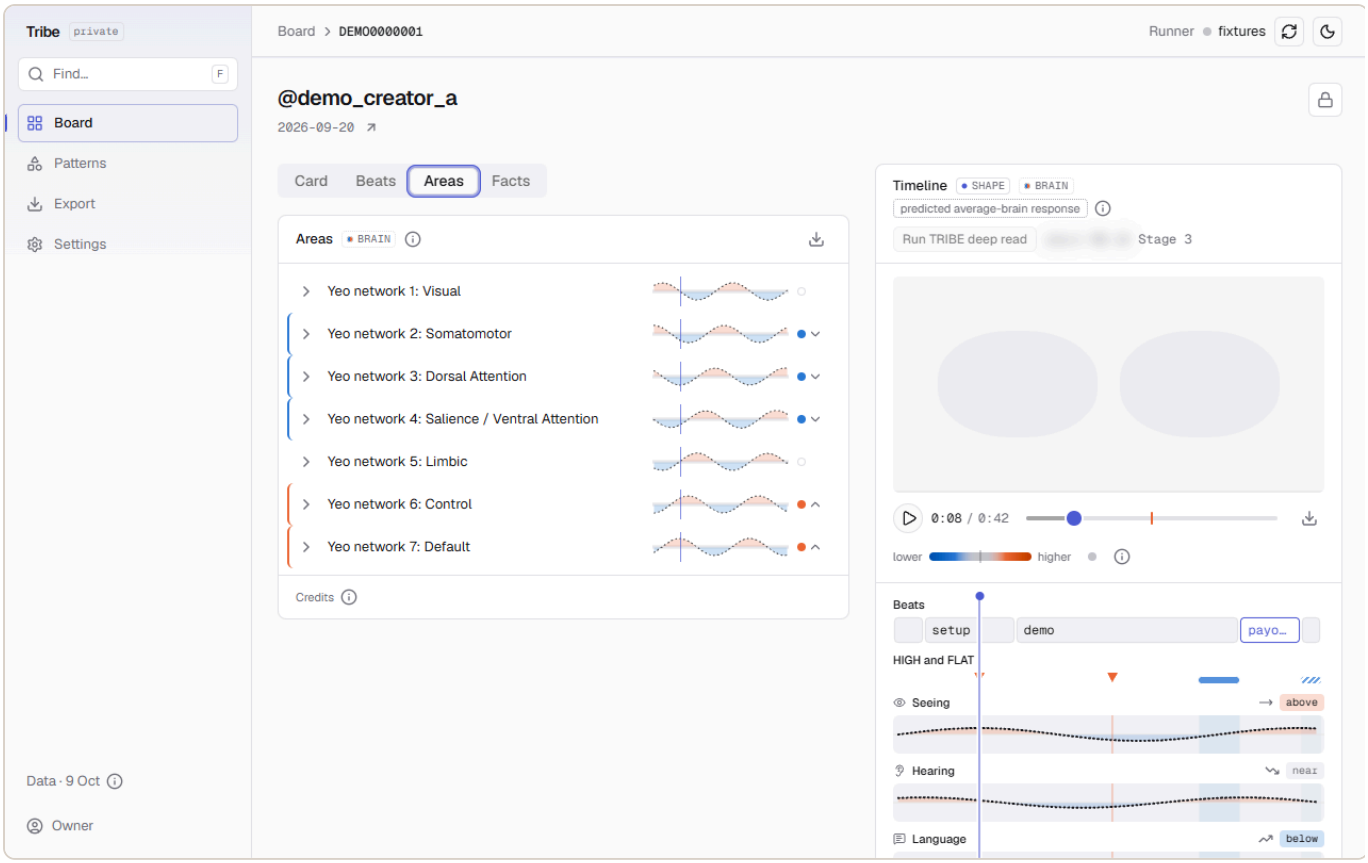
Patterns: how often a shape shows up among outlier reels versus normal ones.

The brain view

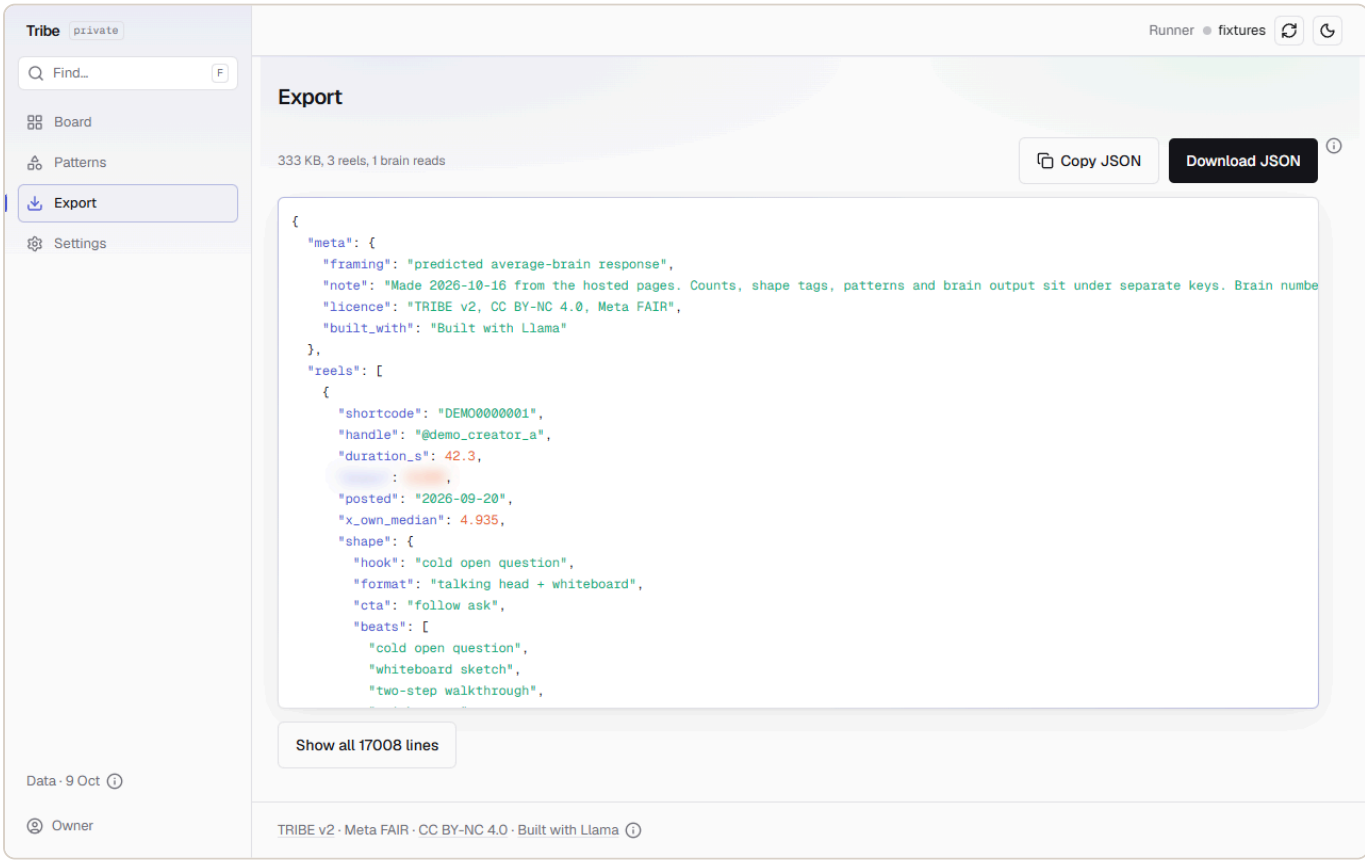


Reel page. The video, timeline and brain share one second. The brain here is a simple demo shape; real reads show a cortical surface.

- It is the **predicted average-brain response**, second by second. It is an average brain, **not your audience**.
- Red and blue mean higher and lower than that brain's own baseline for the clip. They do not mean good and bad.
- There are **no scores**, no ranking of reels by brain response, and no prediction of views, reach or virality. Do not add one.
- Areas appear in a fixed atlas order, never ranked.



Areas tab: seven standard brain networks over time. Open each one to see its finer areas.



Export page: download your board data as JSON. Reel pages also offer CSV downloads of the brain series.

11 Privacy, licence and credits

Privacy

- Keep every key in `.env` and in your hosts' secret settings. Never commit them, print them, or put them in code or screenshots.
- Keep the dashboard behind a login. Do not make the deployment, the video store or the database public.
- Use a read-only database role for the website. Only your own computer holds the full-access one.
- Rotate any token that was ever visible.

Licence and credits

TRIBE v2 is by Meta FAIR (d'Ascoli et al., 2026) and is licensed **CC BY-NC 4.0**: https://creativecommons.org/licenses/by-nc/4.0/ . Model: https://huggingface.co/facebook/tribev2 . Code: https://github.com/facebookresearch/tribev2 . Cite the TRIBE v2 paper as its model card asks.

Built with Llama. The text path uses Llama 3.2 3B under the Llama 3.2 Community License. Llama 3.2 is licensed under the Llama 3.2 Community License, Copyright © Meta Platforms, Inc. All Rights Reserved. Get the weights only from the official Hugging Face page.

Non-commercial only. Allowed: personal use, learning, research. Not allowed: selling access or results, paid tiers, client work, paid deliverables, or using it to win clients. Commercial use needs Meta's permission.

This guide and the tool are not made, endorsed or supported by Meta. Outputs are a predicted average-brain response, not a forecast of views or virality.

Read the Llama 3.2 Community License and Acceptable Use Policy on the Hugging Face page yourself before you start.