

What a fruit fly taught us about making web pages

Kenyon · 2026-09-13 · male CNS connectome v1.0, Janelia + Google Research, CC BY · every figure below is read out of the run files, not typed in

A mushroom body was taken out of a connectome, made to learn, and pointed at web design. It works, in the narrow sense that it does what a fly does. Whether it is the right tool for making pages look better is a different question, and this report answers that one too — with a result that does not flatter the premise.

1. The circuit is real, and it replicates

Every connection is measured: 342 projection neurons onto 2,045 Kenyon cells, the APL feedback cell that reaches 2,044 of them, 17,985 KC → MBON synapses, and the dopamine neurons that teach them. What is *not* measured is named in one place: odours are synthetic, and three gains are tuned by hand. Everything below is 100 runs on both hemispheres, with nothing tuned against the left one.

MEASUREMENT	MEDIAN [5-95%]	HOLDS
KCs active for one odour, %	8.826 [6.45, 10.44]	yes
..with the APL loop removed	52.445 [35.70, 66.96]	yes
preference after 5 punishment trials	-0.246 [-0.28, -0.20]	100% of runs
preference after 5 reward trials	0.300 [0.22, 0.36]	100% of runs
transfer to an odour never presented	0.080 [0.01, 0.37]	yes
..MBON compartments shuffled	-0.096 [-0.18, 0.26]	no
..dopamine mis-aimed	0.005 [-0.08, 0.09]	yes

The one claim that did not survive

A single run suggested that shuffling MBON compartments *reverses* learning. Over 100 shuffles it does not — it destroys it, and only 37% of shuffles reverse anything. The reversal was one seed's luck. This is written into the repository rather than quietly corrected.

2. A web page can be handed to it as if it were a smell

169 detectors over a rendered page — colour, type, space, shape, structure, and the rules a design system enforces — sparsified to the channels an odour would light. The test is not whether the numbers look plausible but whether the resulting code behaves like an odour's.

MEASUREMENT	VALUE
KCs active, synthetic odour	8.7%
KCs active, real web page	9.0%
KC code shared, two pages from one site	0.359
..two pages from different sites	0.119
..example.com/.org/.net, one design on three domains	1.00
..CSS Zen Garden, one document under three designs	0.14

The two controls are what carry it. Three domains serving one design share almost all their Kenyon-cell code; one document under three different designs shares less than two unrelated websites do. The encoder reads the design, not the content.

3. The lesson had to be a pair, not a page

This was the largest single improvement in the project. A single lesson page gives the circuit a point in Kenyon-cell space and no direction to move in. Two pages with the same words and opposite design — one following every rule NebulaKit enforces, one built the way generated design usually is — give it an axis, and an axis is the only thing selection can climb. The rewarded fly's preference went from **+0.05 to +0.34**, which is the magnitude of an odour it was trained on directly rather than a faint transfer.

FINAL PAGE OF	JUDGED BY APPROACH	JUDGED BY AVOID	KC OVERLAP	VS RANDOM
approach	+0.325	-0.256	0.741	100%
avoid	+0.109	-0.067	0.365	7%
naive	+0.208	-0.178	0.565	53%
misaimed0	+0.102	-0.074	0.354	3%
misaimed1	+0.156	-0.103	0.396	7%
misaimed2	+0.115	-0.103	0.400	7%

The twins genuinely oppose: the rewarded fly's page reads +0.325 to one and -0.256 to the other. A page nobody selected scores 0.552 overlap, which is the line every row should be read against.

4. Four things that were wrong, and how they were caught

Correlation flattered a circuit that could not have learned

Scoring the fly by correlating its preference against ratings gave $r = +0.68$ to a fly whose MBON compartments had been shuffled. The magnitude of a preference shift tracks Kenyon-cell overlap with the training set whichever way the lesson pointed, so correlation reports "this resembles things I was taught about" — true of a broken circuit too. The metric is accuracy now.

A single lesioned control is a coin flip

Every lesion in the replication has a distribution that straddles zero. Run once, a lesioned fly built almost exactly what the intact one built, twice in a row, which looked devastating and was a sample of size one. Three of them scatter, and the scatter is the honest picture.

The stand-in rater was invisible to the circuit

A random direction through the 169 detectors produces two classes that are identical in Kenyon-cell space — 0.140 shared code within a class and 0.140 across it. The circuit was being asked about a taste it could not perceive, and failing that test means nothing.

The generator shipped unreadable pages

Left to choose freely, the search produced pages with **39% of their characters below WCAG AA**. Worse, some grounds cannot carry readable text at all: pure black on a 50%-lightness ground is 4.24:1 and white is worse, so no amount of moving the *text* fixes it. The ground has to move first. With that gate in the generator, 16 of 16 random candidates pass.

5. Is this the right way to get the result?

Four methods, the same budget of rendered and measured candidates, the same request: land as close to the reference page as possible. Two objectives are reported for all of them, because scoring a method only on its rival's objective is how a comparison flatters whoever wrote it.

METHOD	DETECTOR COSINE	KC OVERLAP	RENDERS	TIME
rules-only	0.7034	0.490	1	0s
random	0.7645	0.636	320	55s
hill-climb	0.8108	0.534	320	54s
the fly	0.7128	0.741	1,920	—

The uncomfortable result

Each method wins its own objective and neither dominates. A fifteen-line hill-climb with no connectome, no neurons and no learning rule takes the detector-space metric (0.811 against the fly's 0.713) on a sixth of the renders. The fly takes the Kenyon-cell metric (0.741), which is its own objective and not independently shown to mean "better designed".

And a single genome from the gated generator, with *no search at all*, already scores 0.703 and passes every accessibility and performance rule.

Most of the measurable improvement came from encoding the rules, not from the circuit.

6. What can honestly be claimed

- **Claimed:** a circuit taken from a connectome, with nothing added, learns a preference over rendered web pages and applies it to pages it has never seen. Punishment reliably moves the search away from what it was taught — its page shares less code with the lesson than any of thirty unselected pages.
- **Claimed:** the pages it produces are genuinely good on every measurable axis — 0 kB beyond their own HTML, no webfonts, mobile first, and 100% of text clearing WCAG AA. That is the generator's doing.
- **Not claimed:** that the fly makes pages *look better*. No human comparison has been run. Both metrics above measure resemblance to a reference, and resemblance is not beauty.
- **Not claimed:** that this is an efficient way to design. It is not. A direct optimiser beats it on the shared metric at a sixth of the cost.
- **Open:** whether the circuit's objective corresponds to human judgement at all. That needs rated pages and a held-out test, and it is the one experiment that would settle the premise.

7. What would settle it

Forty-three real pages are rendered and waiting; three are rated. With twenty or more rated by hand, held-out prediction becomes answerable: train on some, ask about the rest, and count how often the sign is right. Against stand-in tastes the machinery already reaches 0.615 accuracy where chance is 0.5, with a shuffled-label null at 0.462 and a lesioned circuit at 0.462. The method works; it is waiting on an opinion.

Kenyon · the circuit, the encoder, the generator and this report are all in one repository, and every figure here is regenerated by `python report.py` from the run files rather than transcribed. Connectome: male CNS v1.0, Janelia FlyEM + Google Research, CC BY 4.0. Live: fly.ammoura.me