Teaching a fruit fly's connectome to play chess

Hakim El Ayoubi
Department of Computer Science
University College London

Abstract

A connectome is the complete wiring diagram of a nervous system: every neuron, and every connection between neurons, traced from electron microscope images. The first complete connectome of an adult male fruit fly, published in 2026, records which neurons connect, how many synapses join them and whether each neuron excites or inhibits its targets, but not how strong each connection is in the living animal. I use that gap to ask whether a network with the fly's exact wiring, in which only the unmeasured strengths are learned, can be trained on a task the fly never evolved for, and which of its circuits the trained network relies on. I simulate the 144,209 neurons of the brain, present chess positions through the photoreceptors of the right eye, read the move from the 1,314 descending neurons that normally carry commands from the brain to the body, and train by imitation on 4.3 million positions from human games. Before legality filtering, the trained network's first choice is legal 80% of the time and matches the human move 23% of the time; in matches it beats a random mover and loses almost every game to the weakest Stockfish setting. Silencing groups of neurons costs most for the visual pathway and the descending neurons and nothing measurable for the mushroom body or central complex, but the readout is sensitive enough that small and null effects are hard to interpret. The results are preliminary: no rewired, dense or non-connectome baseline has been trained, the silencing positions overlap the training data, and match scores varied between repeated runs. A final reinforcement-learning phase increased draws and reduced wins, for reasons not yet established.

1 Introduction

A real fruit fly cannot play chess. Each compound eye has fewer than a thousand facets, most of what the fly learns is simple association, and nothing it does requires planning several steps ahead. The question here is narrower: what a network with this fly's wiring can learn when it is trained, which is a different thing from what the fly itself can do.

The connectome used here, released by Janelia, the Cambridge connectomics group and Google Research, covers the male fly's brain and nerve cord [1]. Synapse counts are a reasonable proxy for connection strength but not the same thing: receptor densities, the position of a synapse on the dendrite and other properties invisible to electron microscopy all change how much one neuron influences another. Every cable has been traced, and none of the volume settings were written down.

Chess suits the question for practical reasons. The input is a static image, the output is a discrete choice with an exact legality rule, millions of labelled positions are freely available [4], and strong engines provide a fixed yardstick [10]. The approach follows earlier connectome-constrained models of the fly [3, 9]: wiring and signs are fixed by the data, and only strengths are learned. This report describes one trained network and what can and cannot yet be concluded from it.

Preprint, October 2026. Interactive version: https://github.com/HakimElAyoubi/fly-chess



Figure 1. A frame from the film of the network playing White against Stockfish (Elo 1320 setting). The garden, the fly and the pieces are rendered; each of the fly's moves is computed by the 144,209-neuron model. The panel on the right shows the brain's activity, the board, and the board as it appears on the right eye's columns.

2 Method

2.1 The digital fly

The model contains the 144,209 traced neurons on the brain side of the neck, joined by 21.27 million connections carrying 99.4 million synapses. Each neuron's sign comes from its predicted transmitter: acetylcholine positive (92,375 neurons), glutamate negative (26,874), GABA negative (16,733) and histamine negative (5,902, which include the 4,114 photoreceptors). The 774 dopamine, serotonin and octopamine neurons act through slower modulatory signalling that this model cannot represent, so their outgoing connections are removed; their own rates are still computed. The 1,551 neurons without a confident transmitter prediction are treated as cholinergic.

Each neuron i has a graded rate r_i, set to zero at the start of every move and updated every tick by

$$r_i \leftarrow r_i + k\Big(\tanh\Big(\sum_j W_{ij}\, r_j + I_i + b_i\Big) - r_i\Big), \qquad W_{ij} = \alpha\, \frac{\sigma_j\, n_{ij}}{\sum_{j'} n_{ij'}}, \tag{1}$$

where n_{ij} is the synapse count from j to i, $\sigma_j \in \{+1, -1\}$ the presynaptic sign, I_i the input current (non-zero only for photoreceptors) and b_i a bias. I use $\alpha = 1.5$ and $k = \Delta t/\tau = 0.25$, with 5 ms ticks and a 20 ms time constant; one move is 24 ticks. The raw signed matrix has a leading eigenvalue of 4,089, dominated by the optic lobe's columnar loops, and any single global scale that keeps it stable starves the central brain's weaker pathways; normalising each row avoids this. At initialisation every row of $|W|$ then sums to α, so by the Gershgorin circle theorem no eigenvalue of W has modulus above 1.5. Because $\alpha > 1$, this does not guarantee a stable rest state; the saturating tanh keeps rates bounded, and no run diverged. The bound does not hold after training: the absolute row sums of the trained weights have median 0.93, but 45% of rows exceed 1 and the largest is 1,111.

I chose α from $\{0.5, 0.9, 1.5, 2.5\}$ with two checks taken from fly experiments, so they are calibration rather than independent validation. Driving the 187 labellar sugar-sensing neurons raised the feeding motor neuron MN9 by 1.1% of the network's peak response, which ranks it in the top 2% of all neurons and 4.8 standard deviations above twenty control runs. The controls were meant to be random sensory sets of the same size, but because the pool of other labellar taste neurons has only 58 members, ten of them are the same 58-neuron set; the other ten are random 187-neuron mechanosensory sets. Light on the right eye raised the mean activity of the right L1 and L2 lamina cells to 0.216 of the network's peak response, against 0.0002 on the left.

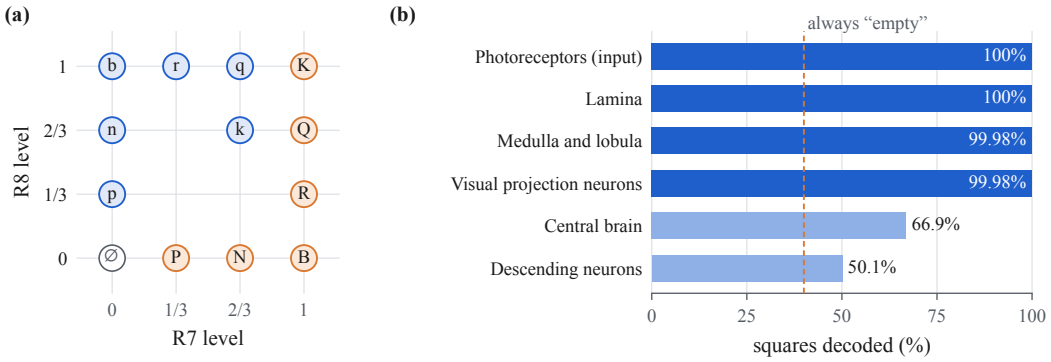


Figure 2. Showing the board to the eye. (a) Each of the thirteen square states is a pair of R7 and R8 levels. ∅ is an empty square. White pieces (upper case) occupy the bottom and right edges, black pieces (lower case) the left and top edges, and the black king the interior point $(\frac{2}{3}, \frac{2}{3})$; the three other interior points are unused. (b) Fraction of squares a linear decoder recovers from each stage of the *untrained* network, on held-out random placements; the dashed line is a decoder that always answers “empty”.

2.2 Input encoding

The connectome assigns most visual neurons to a column of the eye’s hexagonal lattice. Photoreceptors carry no coordinates, so each of the right eye’s 2,239 placed receptors is assigned to the column of its strongest partner among the cell types that occur exactly once per column (L1–L5, Mi1, Mi4, Mi9, Tm1, Tm2, Tm20 and others). The R1–R6 tracing is partial (380 columns) while R7 and R8 cover about 650 to 700, so the board uses the 645 right-eye columns that contain both colour receptors, about ten per square.

Ten columns cannot resolve the shape of a piece, so pieces are encoded by colour (Figure 2). Each of the thirteen square states is a pair of levels $\ell \in \{0, \frac{1}{3}, \frac{2}{3}, 1\}$ on the R7 and R8 channels, presented as currents

$$I = \operatorname{artanh}(0.8\,\ell), \tag{2}$$

which makes the receptors’ rates equally spaced; with a linear drive the two brightest levels were nearly indistinguishable one synapse downstream. The broadband R1–R6 receptors carry the side to move as a uniform brightness shift.

Before any training, I measured how much of the board survives each stage. I ran the untrained model on 8,192 random placements of pieces and 1,024 positions from random play, and fitted a linear decoder per stage; Figure 2b reports held-out random placements. The optic lobe keeps the board essentially intact (99.98% of squares from the medulla and lobula). The central brain yields 66.9% and the descending neurons 50.1%, against 40% for the trivial decoder.

2.3 Training

Each descending neuron’s rate is first standardised with a mean and scale fixed once before training, because the untrained rates at that depth are of order 10^{-5}. A linear readout maps the 1,314 standardised rates to 4,096 from–to logits plus five promotion logits, and a second linear readout predicts the game result (win, draw or loss). Training learns a non-negative gain on every connection and an input and output gain for every neuron, so the effective weight is

$$\tilde{W}_{ij} = g_i^{\text{in}}\, W_{ij}\, e^{\theta_{ij}}\, g_j^{\text{out}}. \tag{3}$$

No term can change a sign or create a connection, since $\tilde{W}_{ij} = 0$ wherever $n_{ij} = 0$. The 21.27 million θ_{ij}, the per-neuron gains, the biases and the readouts make 27.1 million parameters, of which the move readout alone is 5.4 million, trained by backpropagation through all 24 ticks. The loss is the cross-entropy of the move played, plus 0.2 times the cross-entropy of the promotion choice and of

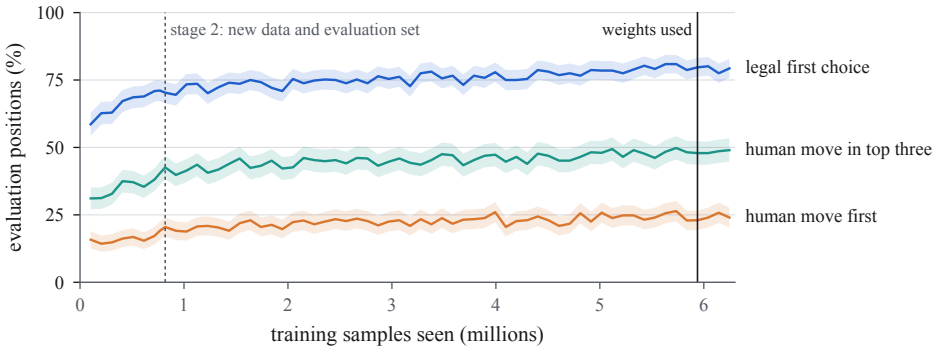


Figure 3. Every evaluation of the supervised run, each on 512 held-out positions, with 95% binomial bands. The dashed line marks the start of stage 2, where both the training data and the evaluation set change; the solid line marks step 11,600, the weights used in the rest of the paper. Legality is measured before filtering; a random from–to pair is legal 0.7% of the time.

the game result. The move head is unmasked during training, so legality has to be learned. A custom autograd function multiplies by a precomputed sparse transpose in the backward pass, so no sparse tensor is ever differentiated.

Data are games between players rated 1600 to 2200 from the Lichess database [4], eight positions per game after the sixth ply, with 10% of games held out for evaluation. Training ran in two stages on four RTX 5090 GPUs with 512 positions per step: 1,500 steps on a warm-up set streamed from August 2026, then 10,700 steps on a larger set; each stage evaluates on its own held-out games, so the evaluation set changes at the boundary (Figure 3). In total training saw 6.2 million samples drawn from 4.3 million distinct positions, in 191 minutes.

2.4 Play and evaluation

In play, scores are restricted to legal moves and an anti-repetition rule forbids moves that recreate an earlier position; without search, the network otherwise shuffles a piece back and forth even from winning positions (three of four games in a pilot). The engine speaks UCI. Matches use four random opening plies, alternating colours and a 300-ply cap scored as a draw. The greedy opponent plays mate in one if available, otherwise the most valuable capture, otherwise a random legal move. Stockfish [10] is used with `UCI_LimitStrength` at its floor of 1320 and 50 ms per move, or at search depth 1.

3 Results

3.1 Playing strength

All results use the weights from step 11,600, which on the stage-2 evaluation set choose a legal move first 79.7% of the time, match the human's move 23.0% of the time and contain it in their top three 47.9% of the time (Figure 3). Over the last 2,000 steps, top-1 agreement moved between 23% and 26%, within the evaluation's sampling error of about ±3.7 points. Table 1 gives match results. Against the Stockfish 1320 setting the network won no games and drew five, all by Stockfish repeating positions. With a score at the boundary, these games do not support a rating estimate, so I do not give one.

3.2 Where the chess happens

I silenced each of 28 groups in turn, clamping their rates to zero at every tick, and measured how often the network chose Stockfish's move on 2,048 positions (Figure 4). These positions are not held

Table 1. Match results for the supervised network. Score counts a win as 1 and a draw as $\frac{1}{2}$; intervals are 95% normal approximations from per-game outcomes, omitted where the score is at the boundary.

Opponent	Games	Won	Drawn	Lost	Score	95% interval
Random mover	400	140	253	7	0.67	0.64–0.69
Greedy capture bot	400	15	216	169	0.31	0.28–0.34
Stockfish, depth 1	200	0	20	180	0.05	0.03–0.07
Stockfish, Elo 1320	200	0	5	195	0.01	–

out: they come from the same deterministic stream as the stage-1 training data, so most of them were trained on, although with the human's move as the label rather than Stockfish's. The intact network agrees with Stockfish 20.3% of the time. Silencing the descending neurons, which removes everything the readout sees, leaves 5.1%, so I treat the 15.2 points between the two as the network's input-dependent performance; one standard error is 0.89 points.

Silencing the photoreceptors, the optic lobe or the visual projection neurons removes 88 to 99% of that margin, and the lamina cells, the Mi, Dm and Tm medulla types and the lobula columnar neurons each remove 35 to 78%. Three visual groups have no measurable effect: the T4/T5 motion detectors (expected for a static board), the medulla tangential cells and the centrifugal neurons. Silencing the 32,160 intrinsic central-brain neurons removes about 60% (20.3% to 11.1%), so the central brain carries much of the signal. None of its best-known structures shows an effect: the mushroom body's Kenyon cells and output neurons, the central complex and the lateral horn each change agreement by less than 0.3 points. That null is weaker evidence than it looks. Kenyon cells receive mostly olfactory input, so a null is expected anatomically; the dopaminergic mushroom-body group cannot have any effect, because its outgoing connections were removed when the graph was built (marked in Figure 4b); and the neuropils that receive the lobula's output were not tested as groups.

The metric has a further weakness. The descending neurons' rates are of order 10^{-5}, and the readout subtracts a fixed mean and multiplies by fixed scales between 10^5 and 10^6. Clamping any group to zero can shift these baselines independently of the information the group carries, which may explain why silencing groups that cannot see the board, such as mechanosensory neurons, still costs about 3 points. Clamping each group to its mean activity, or refitting the readout after silencing, would separate the two effects; I have not done either. A separate probe in the repository gives a consistent picture: a linear readout on a random 1,314-dimensional projection of each population predicts the human move about equally well from the optic lobe (14.2% top-1) and from the descending neurons (14.1%), and slightly worse from the central brain (12.2%).

3.3 What training changed

Training changed the network substantially. Using the full effective weights of (3), the rank correlation with synapse counts over all 21.27 million connections is $\rho = 0.43$. Most of the change sits in the per-neuron gains. Across neurons, the output gains g^{out} have a median of 0.66 and a 1st-to-99th percentile range of 0.07 to 1.78, and the input gains g^{in} range from 0.46 to 2.43 over the same percentiles, with a maximum of 63. The per-connection gains $e^{\theta_{ij}}$ moved less: their median is 0.99, and 41% changed by more than 10%. The project's analysis script reports $\rho = 0.948$ between synapse counts and counts multiplied by the per-connection gains alone; that statistic is uninformative, because multiplying the counts by random gains drawn from the same distribution gives $\rho = 0.953$.

Among the per-connection gains, the inputs turned down hardest come from olfactory receptor neurons, wide-field motion detectors and bristle mechanosensors, none of which carry information about a static board, and 57% of the inputs onto descending neurons were turned down.

Weak connections matter collectively. Keeping only the strongest half of each neuron's inputs, without retraining, changes the chosen move on 41% of 512 positions; keeping the strongest 10% leaves agreement with the full network at 10%. Because each neuron's inputs are normalised, many small

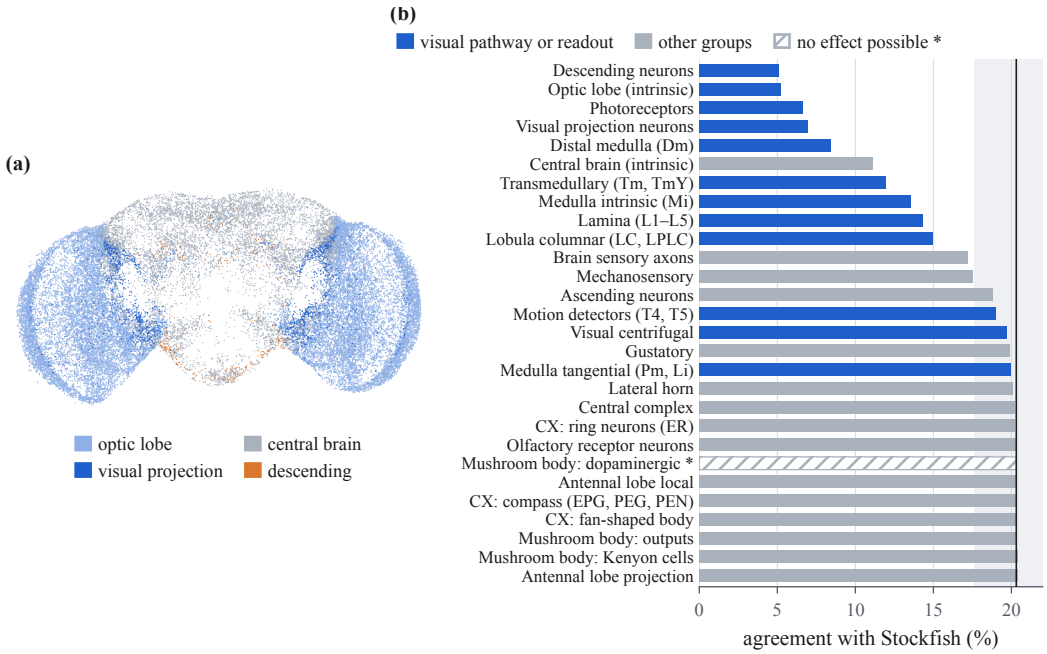


Figure 4. Silencing experiments. (a) Cell-body positions of 30,000 of the 144,209 neurons, front view, coloured by group. (b) Agreement with Stockfish's move after silencing each group, on 2,048 positions that largely overlap the stage-1 training data. Blue: visual pathway or readout; grey: other groups; hatched (*): groups whose outgoing connections are absent from the model, so silencing them cannot have an effect. The vertical line is the intact network (20.3%); the shaded band is ±3 standard errors around it.

connections together make up a large share of its drive.

3.4 Learning from its own games

Imitation never tells the network that a move lost material. In a final phase, it played 20,477 games against the greedy bot and was trained with PPO [8] and generalised advantage estimation [7], with an entropy bonus and a KL penalty towards the frozen supervised policy. The critic is a new scalar head on the descending neurons. This phase trains the per-neuron gains, the biases and the move readout (5.83 million of the 27.1 million parameters); the per-connection gains stay frozen. Each move is rewarded with potential-based shaping, and the final move also receives the result (±1 or 0):

$$F(s, s') = \beta\left(\gamma\,\phi(s') - \phi(s)\right), \qquad \phi(s) = \tanh\!\left(\Delta m(s)/5\right), \tag{4}$$

where Δm is the network's material balance in pawns, $\phi = 0$ for a finished game, $\beta = 0.5$ weights the shaping and $\gamma = 0.995$ is the discount factor. Training games longer than 200 plies are stopped and scored as draws. Other settings: GAE $\lambda = 0.95$, clip ratio 0.2, entropy coefficient 0.01, KL coefficient 0.05, 96 games in parallel and 16 moves per game per iteration. The run took 920 iterations, 1.41 million of the network's own moves and 150 minutes on one RTX 4090.

The weights compared below are the run's best checkpoint by in-training score against the greedy bot, chosen among 92 evaluations, not its final weights; this selection favours the greedy matchup. Replaying the matches with identical settings (Figure 5), the score against the random mover fell from 0.666 to 0.624 (-0.042 ± 0.017), against the greedy bot it rose from 0.307 to 0.326 ($+0.019 \pm 0.019$, within noise), and against Stockfish at depth 1 it was unchanged (-0.009 ± 0.014). Against both weaker opponents, wins and losses fell and draws rose; against the random mover, draws went from 253 to 297 of 400.

The shaping term is one explanation: for a player that cannot calculate, never committing to anything is the cheapest way never to lose material, and the guarantee that potential-based shaping preserves

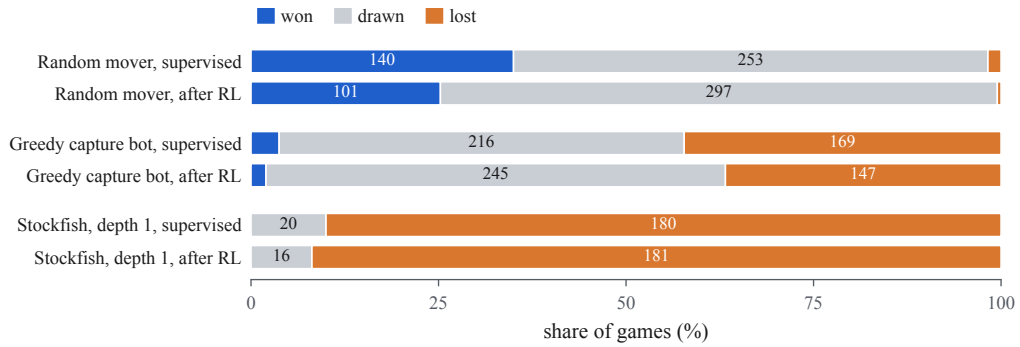


Figure 5. Results before and after reinforcement learning (best in-training checkpoint), with the same match settings and the anti-repetition rule applied to both. Numbers inside bars are game counts. The Stockfish rows use search depth 1; the after-RL Stockfish match stopped after 197 of 200 games.

the optimal policy [6] says nothing about which policy is easiest to reach on a finite budget. Two other explanations are not ruled out. Scoring 200-ply training games as draws also rewards stalling, and the KL anchor and entropy settings were never varied. Separating them needs runs with $\beta = 0$ and without the ply cap, over several seeds, reported at their final checkpoints.

4 Limitations

The results have no reference point yet. I have not trained the same model on a degree-preserving rewired network, with shuffled signs, or as a dense network of matched size, and I have not trained the simpler baselines that would show what the connectome contributes: a readout trained on the frozen untrained brain, or a model reading the photoreceptor encoding directly. Models trained to predict human moves at a given rating [5] give a further comparison for the 23% top-1 figure. Until these exist, nothing here shows that the fly's wiring helps.

The evaluation data are not cleanly separated. The silencing, pruning and compression positions come from the same stream as the stage-1 training data. For silencing, the reported agreements were therefore measured largely on training positions; for pruning and compression, which compare two versions of the network with each other, the overlap matters less.

Match results did not reproduce between runs with the same settings. Against the greedy bot, an earlier 200-game match scored 0.24 (95% interval 0.21–0.28) against 0.31 (0.28–0.34) in the 400-game replay; against Stockfish at depth 1, an earlier match drew 15 of 200 games against 20 in the replay. The earlier matches took over two hours and appear to have run on a laptop CPU, while the replays took minutes on a rented GPU; the readout's sensitivity makes small numerical differences between the two a plausible cause, but I have not tested it by replaying the same games on both.

Other irregularities: the supervised weights (step 11,600) were recovered by hand from a checkpoint corrupted when the compute budget ended mid-write, and the later step-12,200 weights were not recovered; the evaluation during reinforcement learning allowed repeated positions, so Figure 5 uses the replay rather than training-time scores; and every configuration was trained once. The model itself is simple: graded rate units with one time constant rather than spiking cells, no modulatory signalling and no gap junctions, and an eye that sees the pieces and the side to move but not castling rights or en passant.

5 Related work

Shiu et al. [9] simulated the whole female fly brain from its connectome and reproduced feeding and grooming circuits. Lappalainen et al. [3] trained a connectome-constrained model of the visual system on a motion task and found that it predicted measured neural responses; their recipe of fixed

wiring and signs with learned strengths is the one followed here. A few weeks after this work, Cho et al. [2] trained the same male connectome on arithmetic and a small language task and reported that it outperforms degree-preserving random rewirings, which is the control this project still lacks. On the chess side, McIlroy-Young et al. [5] trained networks to predict human moves at specific rating levels, which sets the standard against which move-matching accuracy should be judged.

6 Conclusion and future work

Trained only through its unmeasured strengths, the male connectome learns to choose legal moves and to match human moves about a quarter of the time. Silencing suggests that the visual pathway and the descending neurons carry this ability and that the central brain relays much of it, without implicating any of its named higher-order structures, but the readout's sensitivity and the overlap with training data limit how far those results can be read. The next experiments, in order of cost: the readout-only and direct-encoding baselines; the rewired, sign-shuffled and dense controls; silencing repeated on positions from unseen games with mean-clamping; and reinforcement learning with $\beta = 0$, no ply cap and several seeds. If the connectome then beats its rewired versions, the question I most want to answer is whether the circuits it uses resemble the ones a real fly uses to see.

Code and data availability

Code, trained weights and an interactive atlas of the brain are available at `https://github.com/HakimElAyoubi/fly-chess`. The connectome data are MaleCNS v1.0 [1], released under CC-BY.

Acknowledgements

The connectome data were produced by the Janelia FlyEM Project Team, the Drosophila Connectomics Group at Cambridge and Google Research.

References

[1] Berg et al. Sexual dimorphism in the complete *Drosophila* male central nervous system connectome. *Cell*, 2026. doi: 10.1016/j.cell.2026.08.015.

[2] J. Cho, M. Kang, and D. Kim. A *Drosophila* whole-connectome network can learn human-designed cognitive tasks. *arXiv preprint arXiv:2610.10014*, 2026.

[3] J. K. Lappalainen et al. Connectome-constrained networks predict neural activity across the fly visual system. *Nature*, 2024.

[4] Lichess. Open database of rated games. `https://database.lichess.org`, 2026.

[5] R. McIlroy-Young, S. Sen, J. Kleinberg, and A. Anderson. Aligning superhuman AI with human behavior: chess as a model system. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2020.

[6] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, 1999.

[7] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[9] P. K. Shiu et al. A *Drosophila* computational brain model reveals sensorimotor processing. *Nature*, 2024.

[10] The Stockfish developers. Stockfish chess engine. `https://stockfishchess.org`, 2026.

A Running the model in a browser

The web version of this article runs the trained network on the reader's device. The checkpoint is a 108 MB file of float32 parameters plus the sparse wiring. Three steps reduce it; the first two change nothing beyond floating-point rounding. First, every factor in (3) is folded into one weight per connection; the readout's standardisation of descending-neuron activity is kept separate, because the raw activity of those neurons varies over a range of about 10^{-5} and folding the standardisation into the readout creates large cancelling terms that do not survive rounding. Second, within each row the column indices are sorted, stored as differences from the previous index in a variable-length byte code and compressed, giving 31 MB of wiring. Third, the weights are rounded; Table 2 compares formats against the original model on 1,024 positions.

Table 2. Weight formats compared with the original float32 model on 1,024 positions. Size is the connection weights alone; the 8-bit formats were not shipped and are given uncompressed.

Weight format	Same move	Same top three	Size
float16	100.0%	99.9%	40 MB
8-bit, logarithmic per row	87.4%	91.3%	21 MB
8-bit, linear per row	76.5%	81.2%	21 MB

Half precision reproduces every move. Both 8-bit formats fail for the same reason as the pruning test: a neuron's drive is the sum of many small inputs, which 8 bits per row either rounds to zero or distorts. An 8-bit readout keeps the same move on 98% of positions, below the 99% I required, so it also stays at float16 (10 MB). The page downloads 82 MB once; a background thread decodes it and runs (1) over all neurons for 24 ticks, taking about 1.3 s per move on an Apple M4 laptop. On 64 exported test positions, the browser engine chooses the same move as the original checkpoint in every case.