

IMMORTAL FLY



A continuing digital life.

Exploring persistent digital life through fruit fly brain simulation. An independent research proposal grounded in the MaleCNS connectome.

01

The thesis

What digital continuity means

Immortal Fly investigates persistence for connectome-derived fruit fly simulations: retaining a model's evolving state across interruptions and machine changes.

The engineering objective is a continuing computational trajectory with a traceable lineage. This does not establish biological immortality, consciousness, or the persistence of a subjective identity.

The project publishes this proposal as a research direction. A full persistence engine and experimental results are not yet available.

02

Scientific foundations

From anatomical map to computational model

The MaleCNS connectome is the original scientific resource. The [Cell paper](#) reports 166,691 neurons across the brain and nerve cord. Janelia records the v1.0 dataset release on 8 June 2026 and publication on 3 September 2026.

[Google Research's account](#) identifies HHMI Janelia as project lead. The [official dataset page](#) credits Janelia FlyEM, Cambridge / MRC LMB and Google Research. The dataset is released under CC BY.

Connectivity is a structural constraint; model dynamics require further assumptions. [Shiu et al. \(2024\)](#) studied sensorimotor processing with a computational fly brain model. [Lappalainen et al. \(2024\)](#) combined connectome constraints and task optimization in visual-system models. These are methodological precedents, not evidence of digital immortality.

A filtered simulation graph and an anatomical dataset can have different neuron and connection counts. Their numbers must not be presented as interchangeable.

03

The state we must preserve

Define what a checkpoint contains

A complete checkpoint should identify its graph, model version, parameter set, numeric format and runtime. It should include membrane potentials, refractory counters, learned gains, relevant learning traces, random generator state, the simulation clock and controlled environment state.

State must first pass continuously between simulation windows. Periodic saving cannot restore information that the simulation itself discards at a window boundary.

In the inspected [downstream implementation](#), learned gains already support saving and loading. `FlyBrain.run()` initializes its membrane and refractory arrays per call. This is one implementation-level gap to investigate, not a limitation of the underlying scientific dataset.

04

Recovery & migration

Make continuation inspectable

Write checkpoints atomically with integrity checks and maintain independent backup copies. Record the input history needed for replay, and distinguish committed state from work that may be lost in a crash.

A compatible worker should restore the same checkpoint schema before continuing. One active writer prevents contradictory histories. A deliberate fork receives a new lineage identifier linked to its parent checkpoint.

Hash checks establish file integrity, not consciousness or biological equivalence. Restoring the latest checkpoint may omit uncommitted experience; the record must disclose that gap.

05 The first experiment

A falsifiable continuation test

Run a fixed input sequence in a controlled environment. Preserve a checkpoint, interrupt one worker, restore it and compare its continuation against an uninterrupted reference run.

Measure restored-state agreement, retention of learned changes, the subsequent state trajectory and recovery on a second compatible host. Publish the inputs, runtime, checkpoint schema and comparison procedure.

Cross-platform tests require declared numeric tolerances. A changing live website is not an adequate deterministic test environment. All four tests remain planned; no pass rate or runtime record is claimed here.

06 Status & open questions

What remains to be built

Available today: this proposal and a simplified 384-neuron web model with in-session capture and restore. The original science, public data and downstream code are linked. A durable checkpoint-and-recovery engine for a full connectome remains to be built.

Open questions include state completeness, numerical stability, graph compatibility, recovery gaps, storage cost and how to define a computational lineage when a state is copied.

Immortal Fly is independent of Google Research, HHMI Janelia, Cambridge / MRC LMB and the reference developer. Scientific attribution does not imply affiliation or endorsement.

References & open resources

Original research, public datasets and the downstream implementation.

[01] Berg et al. · Cell · 2026

[The complete male Drosophila CNS connectome](#)

[research.google](#)

[02] HHMI Janelia · Google Research · Cambridge / MRC LMB

[MaleCNS dataset, releases and research credits](#)

[www.janelia.org](#)

[03] Google Research · 03 Sep 2026

[Mapping the complete male fruit fly brain](#)

[research.google](#)

[04] Shiu et al. · Nature · 2024

[A Drosophila computational brain model reveals sensorimotor processing](#)

[www.nature.com](#)

[05] Lappalainen et al. · Nature · 2024

[Connectome-constrained networks predict neural activity across the fly visual system](#)

[www.nature.com](#)

[06] Fruitflydev · inspected revision d450e3f

[Flycoinrh: an independent downstream implementation](#)

[github.com](#)

Immortal Fly is an independent research project. The cited institutions and developers do not endorse it.
Conceptual illustrations are not experimental results.