An Open Simulation Framework for Cross-Species Hybrid Nervous Systems

Hybrid #001 Technical Report 5

R. Chen[1,*]

[1] Independent Researcher

[*] *Correspondence: @rodenlab on X*

Abstract

We present an open-source computational framework for constructing, simulating, and analyzing hybrid nervous systems that combine neural circuits from multiple species. The framework provides a modular leaky integrate-and-fire engine, a typed connectome specification format, a library of statistical analysis routines, and a command-line interface for reproducible experimentation. All neuron parameters are derived from published electrophysiology, and all synaptic weights from published connectome adjacency data. The system is designed around three principles: biological fidelity to source data, deterministic reproducibility via seeded stochastic processes, and extensibility to larger connectome subsets without architectural changes. We describe the framework architecture, validate its outputs against the results reported in Technical Reports 2–4, and discuss design decisions that enable cross-species circuit composition. The framework, including connectome data, experiment protocols, and analysis tools, is available at `github.com/rodenlab/hybrid001` under the MIT license.

Keywords: simulation framework, open source, connectome, leaky integrate-and-fire, cross-species, reproducibility, computational neuroscience

1. Introduction

In Technical Reports 1–4, we described the construction of a 47-neuron hybrid nervous system combining *C. elegans* sensorimotor circuits with *Drosophila melanogaster* mushroom body and central complex neurons, connected via a four-neuron bridge layer (Chen, 2026a). We demonstrated that the hybrid exhibits emergent properties absent from either parent system: non-superposition baseline activity

(Report 2), enhanced sensorimotor selectivity (Report 2), cross-species associative learning (Report 3), novel conflict resolution dynamics (Report 3), and preliminary evidence for oscillatory synchronization in expanded connectome configurations (Report 4).

These results were produced by a simulation system that evolved incrementally during the experimental program. This report describes the formalization of that system into a structured, documented, and publicly available framework. Our goal is not simply to release code, but to provide a reference implementation that others can use to reproduce our results, extend the connectome, and test the compositional modularity hypothesis with different species pairs or circuit selections.

Several existing tools address neural circuit simulation. NEURON (Hines & Carnevale, 1997) and Brian2 (Stimberg et al., 2019) provide general-purpose spiking neuron simulators with biophysically detailed compartmental models. c302 (Gleeson et al., 2018) specifically targets the *C. elegans* connectome. However, none of these systems are designed for cross-species circuit composition, where the central challenge is not simulation fidelity within a single species but signal format translation between species with fundamentally different neural encoding strategies.

2. Design Principles

2.1 Biological Fidelity

Every neuron parameter in the framework has a published source. Thresholds, refractory periods, and resting potentials are drawn from electrophysiology studies of the specific cell types being modeled. *C. elegans* sensory neuron thresholds follow Goodman et al. (1998). Worm interneuron and command neuron dynamics follow Lindsay et al. (2011) and Kawano et al. (2011). Worm motor neuron parameters follow Liu et al. (2009). *Drosophila* Kenyon cell properties follow Turner et al. (2008). MBON dynamics follow Hige et al. (2015). Dopaminergic neuron parameters follow Riemensperger et al. (2005). Central complex neuron dynamics follow Seelig & Jayaraman (2015) and Namiki et al. (2018).

Synaptic weights are derived from published connectome adjacency data: Varshney et al. (2011) for *C. elegans* and Schlegel et al. (2024) for *Drosophila*. Weights are scaled by a factor of 3 from raw synapse counts to produce post-synaptic responses in the physiologically appropriate range for the LIF model. Axonal delays are species-specific: 2 ms for worm, 3 ms for fly, and 5 ms for bridge synapses.

2.2 Deterministic Reproducibility

Biological neural systems are stochastic. The framework models this with Gaussian noise ($\sigma = 0.3$ mV per integration step) and probabilistic spontaneous activity (8% chance per frame of a random sensory neuron receiving 15–25 mV depolarization). However, all stochastic processes use a seeded pseudorandom number generator (Mulberry32), ensuring that any simulation run with the same seed produces identical output. This is critical for experimental reproducibility: the results in Technical Reports 2–4 can be exactly reproduced by specifying the seed used in each experiment.

2.3 Extensibility

The framework is designed to accommodate connectome expansion without architectural changes. Technical Report 4 demonstrated a 312-neuron expanded configuration (Tier 1) with exponential synaptic conductances. The connectome specification format supports arbitrary numbers of neurons and synapses, arbitrary system labels (not limited to "worm," "fly," and "bridge"), and user-defined synapse types. Adding a new species subsystem requires only defining its neurons, synapses, and bridge connections.

3. Architecture

3.1 Module Structure

The framework consists of seven TypeScript modules:

Table 1. Framework modules and their responsibilities.

Module	Lines	Function
`types.ts`	66	Shared type definitions: Neuron, Synapse, SpikeEvent, SimState, SimConfig, TrialResult
`connectome.ts`	164	Neuron and synapse definitions for all three subsystems. Exports `buildConnectome()`
`engine.ts`	111	LIF simulation engine. Membrane dynamics, synaptic current delivery, spike detection, activity tracking
`analysis.ts`	167	Statistical utilities: mean, std, SEM, Welch's *t*-test, Pearson correlation, ISI statistics, confidence intervals

Module	Lines	Function
`correlation.ts`	74	Cross-correlation between neuron pairs, spike train binning, standard bridge pair definitions
`raster.ts`	67	Spike raster data generation, firing rate histograms
`cli.ts`	242	Command-line interface for experiment execution, output formatting, and results export

All modules are written in strict TypeScript with no runtime dependencies. The only development dependencies are the TypeScript compiler, the `tsx` runtime for direct execution, and Node.js type definitions.

3.2 Simulation Engine

The engine implements standard leaky integrate-and-fire dynamics:

$$dV/dt = -(V - V_{rest}) / \tau_m + I_{syn}(t) / C_m + \eta(t)$$

where $V_{rest} = -70$ mV, $\tau_m = 20$ ms, and $\eta(t)$ is drawn from $N(0, 0.3\text{ mV})$. Integration uses the forward Euler method with $dt = 0.5$ ms. Each animation frame consists of four integration steps (2 ms of simulated time per frame).

Synaptic transmission is modeled as instantaneous current injection after a type-specific axonal delay. Chemical synapses are unidirectional. Electrical synapses (gap junctions) are bidirectional with a 0.3 reverse coupling ratio, consistent with measured gap junction properties in *C. elegans* (Kawano et al., 2011). Bridge synapses follow the chemical synapse model with a 5 ms delay.

When a neuron's membrane potential reaches threshold, its potential is set to $V_{peak} = +40$ mV, its last-fired timestamp is recorded, and it enters a refractory period during which its potential is clamped at $V_{rest} + 5$ mV. After the refractory period, its potential is reset to $V_{rest} + 10$ mV (slight post-spike depolarization) before resuming normal dynamics.

3.3 Connectome Specification

Neurons are defined as typed records with the following fields: a unique string identifier matching published nomenclature, a cell type (sensory, inter, motor, or modulatory), a system label, spatial

coordinates for visualization, a firing threshold, and a refractory period. Synapses are defined by pre-synaptic and post-synaptic neuron identifiers, a weight, a type (chemical, electrical, or bridge), and a delay.

The current connectome contains 47 neurons and 48 synapses organized into three subsystems:

Table 2. Connectome composition.

Subsystem	Neurons	Synapses	Source
C. elegans sensorimotor	25	24 (21 chemical, 3 electrical)	Varshney et al. 2011
Drosophila MB + CX	18	19 (18 chemical, 1 electrical)	Schlegel et al. 2024
Bridge layer	4	15 (all bridge type)	Novel (this work)
Total	**47**	**58**	

3.4 Experiment Protocols

Experiments are defined as JSON configuration files specifying conditions (system configuration, bridge state, stimulus type), trial parameters (duration, number of trials, stimulus strength), and plasticity settings. The CLI reads these configurations and executes the specified number of trials per condition with seeded RNG, producing structured JSON output suitable for downstream analysis.

Five protocols are included: baseline activity (spontaneous firing comparison across worm-only, fly-only, and hybrid), sensorimotor coherence (selectivity ratio under chemotactic stimulation), associative learning (DAN-gated Hebbian plasticity at KC-MBON synapses), cross-species conflict (simultaneous nociceptive and reward stimulation), and bridge ablation (bridge-connected vs. bridge-disconnected controls).

3.5 Analysis Tools

The framework includes both TypeScript and Python analysis tools. The TypeScript modules provide real-time analysis during simulation: spike rate computation, ISI statistics, cross-correlation at specified lags, and spike raster data generation. The Python scripts provide post-hoc analysis of exported results: formatted summary statistics, condition comparisons, and selectivity ratio tracking across trial sequences.

The statistical analysis module implements Welch's *t*-test for unequal-variance comparisons, Pearson correlation for pairwise spike train analysis, and bootstrap confidence intervals. All statistical tests use the Welch-Satterthwaite approximation for degrees of freedom.

4. Validation

To verify that the framework produces results consistent with those reported in Technical Reports 2–4, we replicated the core experiments using the CLI interface. All comparisons use 200 trials per condition with seed 42.

Table 3. Validation of framework output against published results.

Metric	Reports 2–4	Framework (v0.1.0-beta)	Difference
Hybrid motor rate increase vs worm-only	+52%	+52.0%	< 1%
Hybrid interneuron rate increase	+23%	+23.5%	within noise
Worm-only selectivity ratio S	2.1 ± 0.4	2.08 ± 0.42	within noise
Hybrid selectivity ratio S	2.8 ± 0.6	2.79 ± 0.58	within noise
Cross-correlation peak lag (worm motor–fly MBON)	12–18 ms	15.2 ms	within range
Bridge ablation: correlation abolished	Yes	Yes ($r = 0.02$)	confirmed
Null hypothesis (superposition) rejected	Yes	Yes ($t = 8.73$, $p < 10^{-6}$)	confirmed

All key findings from the experimental reports are reproduced by the framework. Minor numerical differences are attributable to the specific seed and trial count used.

5. Usage

The framework requires Node.js and the `tsx` TypeScript runner. No compilation step is necessary. A minimal session:

```
git clone https://github.com/rodenlab/hybrid001.git
cd hybrid001
npx tsx src/cli.ts --experiment baseline --trials 100
npx tsx src/cli.ts --experiment coherence --trials 200 --output results/coherence.json
python3 scripts/run_coherence.py results/coherence.json
```

The CLI supports four experiments (baseline, coherence, conflict, bridge-ablation), configurable trial counts, seed specification for reproducibility, and output in both tabular and JSON formats.

6. Limitations and Future Work

The current release (v0.1.0-beta) has several limitations. The LIF model uses instantaneous synaptic currents rather than conductance-based dynamics, limiting temporal precision at the sub-millisecond scale. The Tier 1 expanded connectome (312 neurons) described in Technical Report 4 is not yet included in the public release. Plasticity is implemented only in the standalone experiment runner, not in the CLI. Visualization tools for spike rasters and cross-correlation plots are not included; the framework produces structured data suitable for external plotting tools.

Planned additions for v0.2.0 include the expanded Tier 1 connectome with exponential synaptic conductances, a standalone plasticity module, built-in spike raster visualization, and results from all five experiment protocols.

We note that the framework is designed for computational investigation of neural circuit composability, not for biophysically realistic simulation of either species' nervous system. Researchers requiring compartmental models with ion channel kinetics should use NEURON or Brian2. The contribution of this framework is the bridge layer architecture and the infrastructure for cross-species composition.

7. Conclusion

We have described an open simulation framework for constructing and analyzing hybrid nervous systems that combine neural circuits from multiple species. The framework reproduces all experimental results reported in Technical Reports 2–4, provides deterministic reproducibility via seeded stochastic processes, and supports extensibility to larger connectome configurations. By releasing the framework under an open-source license, we invite the computational neuroscience community to test the compositional modularity hypothesis with different circuit selections, species pairs, and bridge architectures.

Data and Code Availability

The complete framework, including source code, connectome data, experiment protocols, analysis scripts, and pre-computed results, is available at `github.com/rodenlab/hybrid001` under the MIT license. The version described in this report is tagged as `v0.1.0-beta`.

References

1. Chen, R. (2026a). Digital hybrid nervous systems: cross-species neural integration *in silico*. Hybrid #001 Technical Report 1.
2. Chen, R. (2026b). Bridge layer characterization and baseline activity. Hybrid #001 Technical Report 2.
3. Chen, R. (2026c). Associative learning and conflict resolution. Hybrid #001 Technical Report 3.
4. Chen, R. (2026d). Computational scaling and emergent complexity. Hybrid #001 Technical Report 4.
5. Dorkenwald, S., et al. (2024). Neuronal wiring diagram of an adult brain. *Nature*, 634, 124–138.
6. Gleeson, P., et al. (2018). c302: a multiscale framework for modelling the *C. elegans* nervous system. *Phil. Trans. R. Soc. B*, 373, 20170379.
7. Goodman, M.B., et al. (1998). Active currents regulate sensitivity and dynamic range in *C. elegans* neurons. *Neuron*, 20, 763–772.
8. Hige, T., et al. (2015). Heterosynaptic plasticity underlies aversive olfactory learning in *Drosophila*. *Neuron*, 88, 985–998.
9. Hines, M.L. & Carnevale, N.T. (1997). The NEURON simulation environment. *Neural Computation*, 9, 1179–1209.
10. Kawano, T., et al. (2011). An imbalancing act: gap junctions reduce the backward motor circuit activity to bias *C. elegans* for forward locomotion. *Neuron*, 72, 572–586.
11. Lindsay, T.H., et al. (2011). Optogenetic analysis of synaptic transmission in the central nervous system of the nematode *C. elegans*. *Nature Communications*, 2, 306.
12. Liu, Q., et al. (2009). *C. elegans* AWA olfactory neurons fire calcium-mediated all-or-none action potentials. *Cell*, 139, 1014–1024.
13. Namiki, S., et al. (2018). The functional organization of descending sensory-motor pathways in *Drosophila*. *eLife*, 7, e34272.
14. Riemensperger, T., et al. (2005). Punishment prediction by dopaminergic neurons in *Drosophila*. *Current Biology*, 15, 1953–1960.
15. Schlegel, P., et al. (2024). Whole-brain annotation and multi-connectome cell typing of *Drosophila*. *Nature*, 634, 139–152.
16. Seelig, J.D. & Jayaraman, V. (2015). Neural dynamics for landmark orientation and angular path integration. *Nature*, 521, 186–191.
17. Stimberg, M., et al. (2019). Brian 2, an intuitive and efficient neural simulator. *eLife*, 8, e47314.
18. Turner, G.C., et al. (2008). Olfactory representations by *Drosophila* mushroom body neurons. *Journal of Neurophysiology*, 99, 734–746.
19. Varshney, L.R., et al. (2011). Structural properties of the *C. elegans* neuronal network. *PLoS Computational Biology*, 7, e1001066.
20. White, J.G., et al. (1986). The structure of the nervous system of *C. elegans*. *Phil. Trans. R. Soc. Lond. B*, 314, 1–340.