

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 WHOLE-BRAIN CONNECTOMIC GRAPH NEURAL NET- WORK ENABLES WHOLE-BODY LOCOMOTION CON- TROL IN DROSOPHILA

Anonymous authors

Paper under double-blind review

ABSTRACT

Whole-brain connectome provides a structural blueprint for linking neural circuits to behavior, yet its application to embodied control remains largely unexplored. We introduce the fly-connectomic Graph Neural Network (flyGNN), a reinforcement learning controller whose architecture is instantiated directly from a complete adult Drosophila connectome. Our flyGNN models the connectome as a directed message-passing graph, partitioned into afferent, intrinsic, and efferent pathways that structure information flow from sensory inputs to motor outputs. Integrated with a dynamically controllable biomechanical model of Drosophila, flyGNN achieves stable control across diverse locomotion tasks, including gait initiation, walking, turning, and flight, without task-specific architectural tuning. These results demonstrate that whole-brain connectivity can directly support embodied reinforcement learning, establishing a new paradigm for connectome-based control algorithms.

1 INTRODUCTION

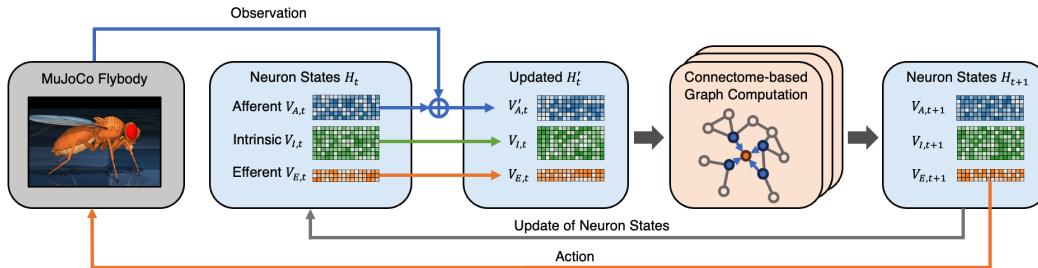


Figure 1: **Overview of the flyGNN enabled whole-body locomotion control framework.** Observations are mapped into afferent neuron states through dimensional transformation. Neural states are then propagated through the connectome-based graph computation module, where update rules are directly constrained by the FlyWire connectome. The updated efferent neuron states are converted into actions to drive the whole-body locomotion of an embodied drosophila model in MuJoCo. (An anonymous webpage with demonstration videos and source code is available at: <https://sites.google.com/view/flygnn>).

Understanding how neural circuits give rise to behavior is a long-standing challenge shared by neuroscience and artificial intelligence. Recent advances in whole-brain connectomics have provided open-source neuronal wiring diagrams of the adult Drosophila brain at synaptic resolution (Dorkenwald et al., 2024). These resources enable the possibility of linking complete brain structure to the sensorimotor control of a physical body. Yet, a fundamental challenge remains: how can static connectomes be transformed into dynamic, functional models that reproduce the intricate and adaptive motor behaviors of animals? Answering this question requires bridging two active lines of research: (i) mechanistic modeling that leverages whole-brain connectivity, and (ii) learning frameworks that generate high-dimensional whole-body movements.

On the control side, reinforcement learning has produced controllers that can drive humanoids, quadrupeds, and musculoskeletal agents to achieve challenging tasks. However, these policies typically rely on generic multilayer perceptrons (MLPs) or hand-crafted modules. While these architectures are useful, they bear little relevance to the biological structure of the brain, making it difficult to align them with real nervous systems. On the connectomics side, models constrained by circuit anatomy have provided insights into sensory and premotor computations (Azevedo et al., 2024), yet most efforts remain limited to specific subsystems and simplified behaviors. What is missing is a whole-brain, embodied approach that respects the connectivity blueprint and operate in a closed-loop physical environment.

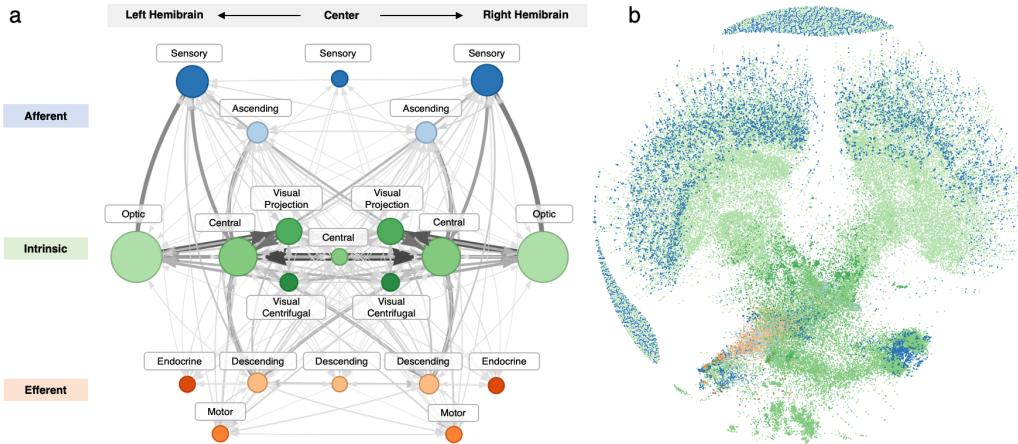


Figure 2: Structure of the fly-connectomic Graph Neural Network. (a) Aggregated synapse graph of the fly connectome, grouped into afferent (blue), intrinsic (green), and efferent (orange) sets across left hemibrain, central, and right hemibrain compartments. Node sizes reflect the number of neurons in each group, and arrows indicate the direction and relative strength of connectivity. (b) Force-directed graph layout (Kobourov, 2012) of the same neural network. The spatial layout reveals hemispheric symmetry and functional clustering.

Here, we bridge these domains by introducing the fly-connectomic Graph Neural Network (fly-GNN), a controller whose architecture is directly adapted from the Drosophila whole-brain connectome. We treat the connectome as an unweighted, directed graph and partition nodes into afferent, intrinsic, and efferent sets following the information flow directions. At each control step, flyGNN takes the sensory signals as input through afferent pathways, propagates the information through the brain-wide graph via message-passing, and outputs motor actions through the efferent pathways to actuate the body.

We evaluate flyGNN on a physics-based Drosophila model (Vaxenburg et al., 2025), demonstrating that a single connectome-grounded policy class supports multiple locomotor behaviors, including gait initiation, straight walking, turning, and flight. Beyond behavioral performance, we analyze neural representations with low-dimensional embeddings and the large-scale visualizations reveal emergent functional segregation across sensory, central, and motor populations. This differentiation emerges solely from the connectome topology without biological priors, suggesting that structural wiring may induce functional specialization.

In this work, we introduce fly-connectomic Graph Neural Network (flyGNN), a neural network controller whose architecture is directly adapted from a Drosophila whole-brain connectome. We show that this connectome-structured network can drive diverse locomotor behaviors in a physics-based fly simulation, including gait initiation, walking, turning, and flight. By demonstrating neural control of movement directly from the connectome, this work establishes a framework for studying how whole-brain neural architectures support whole-body movement behavior, advancing both neuroscience and embodied artificial intelligence.

108 **2 RELATED WORK**

110 **Connectomics-based neural network modeling.** Advances in connectomics have deepened our
 111 understanding of circuit-level organization in central nervous systems of animal models. In partic-
 112 ular, the FlyWire project provides a whole-brain reconstruction of *Drosophila* at synaptic resolu-
 113 tion, offering the structural basis for modeling complete neural dynamics (Dorkenwald et al., 2024;
 114 Schlegel et al., 2024; Zheng et al., 2018). Prior works have highlighted the explanatory power of
 115 such data in restricted domains: ventral nerve cord reconstructions revealed leg-wing coordination
 116 circuits (Azevedo et al., 2024; Lesser et al., 2024), and models of motor neurons have been ap-
 117 plied to feeding or grooming behaviors (Shiu et al., 2024). Connectome-constrained networks have
 118 also been used to predict neural activity in the visual system (Lappalainen et al., 2024). However,
 119 these approaches often focus on specific subsystems or tasks, leaving open the question whether
 120 whole-brain connectivity can generate realistic control of embodied locomotion behaviors.

121 **Embodied movement control.** In parallel, embodied intelligence research has advanced locomo-
 122 tion in simulated humanoids (Kumar et al., 2021; Cheng et al., 2024), quadrupeds (Ding et al.,
 123 2021), and musculoskeletal agents (He et al., 2024; Wei et al., 2025) using reinforcement learning.
 124 In the *Drosophila* domain, physics-based models such as NeuroMechFly (Lobato-Rios et al., 2022;
 125 Wang-Chen et al., 2024) and flybody (Vaxenburg et al., 2025) have enabled detailed simulations of
 126 walking and flight in MuJoCo (Todorov et al., 2012). Yet, controllers for these systems are typically
 127 built from generic MLPs or manually designed central pattern generators, lacking direct biological
 128 grounding. This limits both interpretability and the ability to connect neural structure to behavior.
 129 Our work differs by directly embedding the connectome into the controller architecture, combining
 130 embodied simulation with structural priors to study both performance and neural representation.

131 **3 METHOD**

132 **3.1 CONNECTOME-STRUCTURED NEURAL NETWORK ARCHITECTURE**

133 We consider the problem of embodied sensorimotor control for a virtual fruit fly agent interacting
 134 with a physics-based environment provided by flybody. Let the state at time step t be denoted by
 135 $s_t \in \mathcal{S}$. The agent receives an observation $x_t \in \mathbb{R}^{d_{\text{in}}}$, which corresponds to a set of processed
 136 features including proprioceptive and exteroceptive signals during movement and environmental
 137 interaction. Based on this input, the neural controller produces an action $a_t \in \mathbb{R}^{d_{\text{out}}}$, representing the
 138 motor outputs that drive the flybody model to perform locomotion behaviors.

139 The controller is parameterized by a graph neural network whose architecture follows the anatomical
 140 connectivity of the *Drosophila* brain. Specifically, the connectome is implemented as a directed
 141 graph $G = (V, E)$, where each node $v \in V$ corresponds to a neuron and each directed edge $(u, v) \in$
 142 E indicates the existence of a synaptic connection from neuron u to neuron v . We simplified further
 143 biological details (such as neurotransmitter types, synapse counts, or cell morphology) and modeled
 144 the connectome to capture the existence and direction of synaptic connections.

145 Following FlyWire’s classification of flow types, we partitioned the nodes into three disjoint sets:

- 146 **Afferent neurons:** $V_a \subset V$, which receive external sensory inputs,
- 147 **Intrinsic neurons:** $V_i \subset V$, which mediate signals within the network, and
- 148 **Efferent neurons:** $V_e \subset V$, which produce motor outputs to the body model.

149 Our flyGNN architecture implements the complete *Drosophila* connectome through a graph neural
 150 network where each of the 139,246 neurons (comprising 19,262 afferent, 118,496 intrinsic, and
 151 1,488 efferent neurons) is represented by node embeddings. The architecture’s biological fidelity
 152 and substantial capacity support complex locomotion capabilities while maintaining the structure of
 153 the connectome.

154 Thus, $V = V_a \cup V_i \cup V_e$ with $V_a \cap V_i \cap V_e = \emptyset$. Each neuron $v \in V$ is associated with a latent state
 155 vector $h_v \in \mathbb{R}^C$, and we collected all neuron states at time t as a matrix:

156
$$H_t \in \mathbb{R}^{|V| \times C} \quad (1)$$

162 At each step, the sensory input x_t was first embedded into afferent states by an encoder layer Enc_θ :

$$164 \quad z_t = \text{Enc}_\theta(x_t) \in \mathbb{R}^{|V_a| \times C} \quad (2)$$

165

166 The afferent states were then updated by a gated mechanism that combined the encoded sensory
167 features and the previous afferent states:

$$168 \quad h'_a = \tanh\left(W_g [z_t \parallel H_t[V_a]] + b_g\right), \quad a \in V_a, \quad (3)$$

170

171 where $[\cdot \parallel \cdot]$ denotes concatenation and W_g, b_g are learnable parameters. This update ensures that
172 sensory information is integrated with the existing neural states before message-passing.

173 After afferent updates, the complete node states H_t were propagated through the connectome graph
174 by a message-passing operator $\text{MP}(\cdot)$. In the simplest case of a Graph Convolutional Network
175 (GCN) (Kipf & Welling, 2017), the update rule could be formulated as the following:

$$176 \quad H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right) \quad (4)$$

178

179 where $\tilde{A} = A + I$ is the adjacency matrix with self-loops, \tilde{D} is the diagonal degree matrix, $W^{(l)}$ is
180 a learnable weight matrix, and $\sigma(\cdot)$ is a nonlinear activation. This operator propagated neural infor-
181 mation along the edges of the connectome, producing updated states for both intrinsic and efferent
182 neurons. In practice, the message-passing module can be instantiated not only as a GCN but also
183 with more expressive architectures such as GraphSAGE (Hamilton et al., 2018), GAT (Veličković
184 et al., 2018), or PNA (Corso et al., 2020), which offered improved performance and scalability (See
185 Appendix C for comparative experiments).

186

187 The updated efferent states $H_t[V_e]$ were flattened and passed through a decoder layer Dec_ϕ :

$$188 \quad a_t = \text{Dec}_\phi\left(H_t[V_e]\right) \quad (5)$$

189

190 This step maps the neural features of the efferent neurons directly into continuous motor outputs,
191 denoted as a_t . These outputs served as motor commands to actuate the flybody, a biomechanical
192 model of Drosophila implemented in MuJoCo (Vaxenburg et al., 2025). The agent then updated the
193 physical state of the fly and produced the next sensory observation x_{t+1} , making the sensorimotor
194 in a closed-loop manner. Over a trajectory $\tau = (x_0, a_0, x_1, \dots, x_T)$, the objective of the fly agent
195 was to generate actions that realize stable and efficient locomotion.

196

197 In summary, Algorithm 1 outlines the complete forward computation of our neural control policy
198 which follows the anatomical connectivity of the Drosophila brain - from sensory input through
199 afferent gating, message-passing across the connectome, to efferent decoding into motor actions.

200 **Algorithm 1:** fly-connectomic Graph Neural Network (flyGNN)

201 **Input:** Sensory input x_t , connectome graph $G = (V, E)$ with node partitions V_a, V_i, V_e ;
202 encoder Enc ; gate (W_g, b_g) ; message-passing MP ; decoder Dec

203 **Output:** Motor output a_t

204 **for** each time step t **do**

205 $z_t \leftarrow \text{Enc}(x_t)$;

206 $H_t[V_a] \leftarrow \tanh(W_g [z_t, H_t[V_a]] + b_g)$;

207 $H_{t+1} \leftarrow \text{MP}(H_t, E)$;

208 $a_t \leftarrow \text{Dec}(H_{t+1}[V_e])$;

209 Apply a_t to MuJoCo to obtain x_{t+1} ;

210 **end**

211

212

213 3.2 TRAINING PIPELINE

214

215 Our training pipeline consisted of two stages: first, we initialized the connectome-based policy using
imitation learning from expert trajectories, and second, we fine-tuned the model with reinforcement

learning to directly optimize for task rewards. This two-stage design leveraged demonstration data for rapid initialization while preserving the capability for adaptive policy improvement.

To provide an initial policy, we collected expert trajectories by rolling out an MLP-based policy for the flybody which was originally trained with imitation learning to generate high-quality demonstrations of locomotion. We used these trajectories to train our connectome-based model by imitating the expert’s action distributions.

Specifically, the policy predicted Gaussian parameters (μ_s, σ_s) given the same observations as the expert, and was optimized to minimize a loss combining Kullback–Leibler divergence with an annealed mean squared error (MSE) regularizer:

$$\mathcal{L}_t = D_{\text{KL}}(\mathcal{N}(\mu_t, \sigma_t^2) \parallel \mathcal{N}(\mu_s, \sigma_s^2)) + \lambda(t) \left(\|\mu_s - \mu_t\|_2^2 + \alpha \|\log \sigma_s - \log \sigma_t\|_2^2 \right) \quad (6)$$

where α is a constant to balance the scale of μ_s and $\log \sigma_s$, and $\lambda(t)$ decreases during training so that distributional matching dominates in later stages. This procedure initialized the model with stable behaviors for walking and flight tasks.

After initialization, we fine-tuned the connectome-structured policy using Proximal Policy Optimization (PPO) to enable direct learning from rewards. For value estimation, we used a simple MLP as the value network. The environments in MuJoCo were adapted into gym-like interfaces with parallel rollouts to increase throughput, and distributed training with Distributed Data Parallel (DDP) was used for scalability. The policy was updated following the clipped surrogate objective with value and entropy regularization:

$$\mathcal{L}_{\text{PPO}} = \mathbb{E}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) - c_v (V_\theta(s_t) - R_t)^2 + c_e \mathcal{H}[\pi_\theta(\cdot | s_t)] \right] \quad (7)$$

Where $r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)}$ is the probability ratio between new and old policies, \hat{A}_t is the GAE advantage, R_t is the return, and \mathcal{H} is the entropy bonus. This stage allowed the model to refine beyond demonstration data and adapt to task-specific dynamics, while retaining the inductive bias imposed by the connectome architecture.

4 EXPERIMENTS

We evaluated flyGNN on four locomotor tasks: gait initiation, straight walking, turning, and flight within the flybody physics simulator (Vaxenburg et al., 2025). The environments followed the default flybody setup, with binocular visual signals added to the original sensory inputs in the walking tasks. This embodied setting provided diverse multimodal feedback, including proprioception, mechanosensation, and vision, enabling us to test whether a connectome-structured network could flexibly generate stable control policies across different modes of movement. Detailed training process and parameters are provided in the Appendix A and detailed observation and action definitions are provided in Appendix B. The demonstration videos for each task are available on our project webpage.

The flyGNN controller was evaluated under two primary task configurations. For walking-related tasks (gait initiation, straight walking, and turning), the model processed a high-dimensional observation space of 1,253 features, integrating proprioceptive, exteroceptive, and augmented visual inputs. It subsequently output continuous actions of 59 dimensions to precisely control joint actuators and adhesion mechanisms. For the flight task, an observation space of 104 features was implemented, focusing on the core proprioceptive and kinematic states, which generated 12-dimensional actions to modulate wing torques, body joints and a wing pattern generator. The training process for both configurations adhered to the pipeline mentioned above.

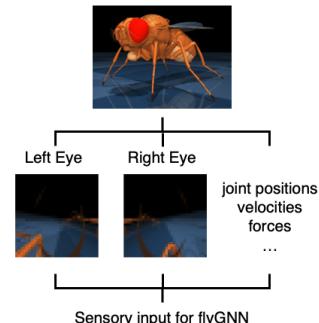
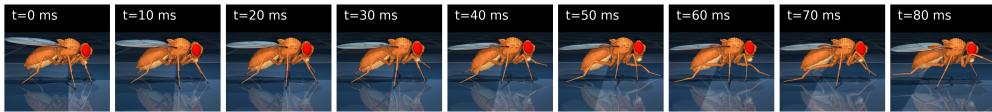


Figure 3: **Demonstration of sensory input for locomotion tasks.** Visual inputs from both eyes are integrated with other sensations to form the sensory input.

270
271 **Gait Initiation**

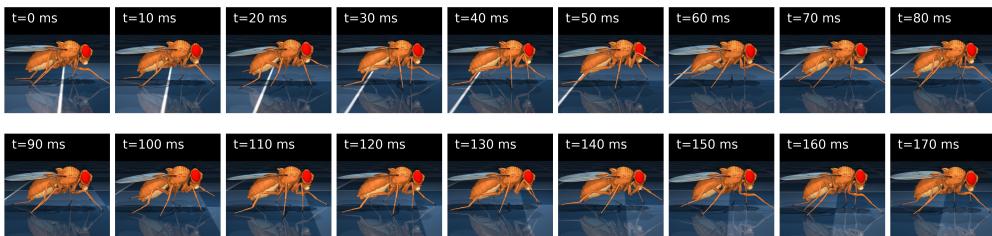
272 Building on the training pipeline described above, we evaluated flyGNN on a walking task at a target
 273 velocity of 3 cm/s. We first examined the process of gait initiation, focusing on the transition from
 274 rest to the onset of stable locomotion. Figure 4 illustrates snapshots of the simulated fly prior to
 275 the first complete gait cycle. The initiation phase lasted for roughly the first 80 ms, during which
 276 irregular steps gradually gave way to rhythmic and coordinated leg movements.



277
278
279
280
281
282 **Figure 4: Gait initiation dynamics.** Snapshots of the simulated fly during the onset of locomotion,
 283 prior to the first complete gait cycle. The sequence illustrates how the agent transitions from rest
 284 into stepping, with irregular and asymmetric leg movements gradually giving rise to a coordinated
 285 pattern.

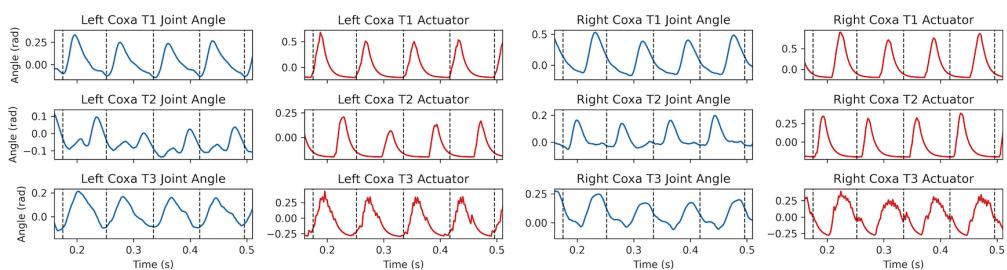
286
287 **Straight-Line Walking**

288 Following the analysis of gait initiation, we next evaluated flyGNN on a straight-line walking task
 289 at a target velocity of 3 cm/s. As illustrated in Figure 5, the model produced stable forward locomotion
 290 with clear tripod coordination. The simulated fly maintained a consistent body trajectory over
 291 hundreds of milliseconds, without exhibiting drift or collapse, indicating that the learned controller
 292 generalizes well to sustained walking.



293
294
295
296
297
298
299
300
301
302 **Figure 5: Walking dynamics.** Snapshots of the simulated fly walking in a straight line at a velocity
 303 of 3 cm/s. The model maintains stable stepping sequences with tripod coordination emerging natu-
 304 rally from flyGNN policy model.

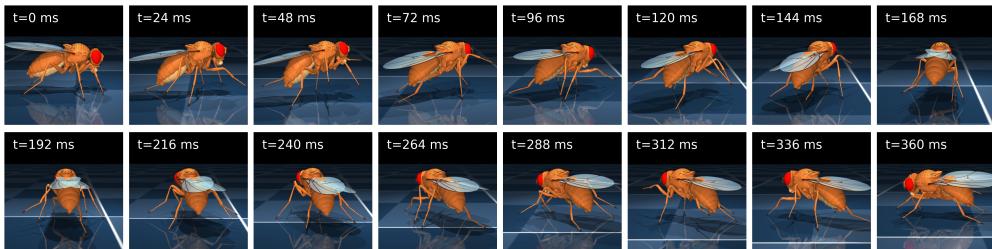
305 Joint-level analysis in Figure 6 shows that actuator outputs are tightly coupled with kinematic tra-
 306 jectories: contralateral legs alternate in phase, producing the classical tripod gait pattern seen in
 307 Drosophila. These results demonstrate that flyGNN is sufficient to generate stable straight walking
 308 once locomotion is initiated.



310
311
312
313
314
315
316
317
318
319
320
321
322
323 **Figure 6: Joint kinematics and actuator activations during walking.** We visualize the angles
 324 (blue) and actuator activations (red) of coxa joints from left (T1–T3) and right (T1–T3) legs over
 325 multiple gait cycles. Dashed vertical lines mark gait phases based on the troughs of the left T1 coxa.
 326 The model reproduces alternating tripod-like coordination, with left T1/T3 synchronized with right
 327 T2, and left T2 synchronized with right T1/T3, consistent with expected fly walking patterns.

324
325
Turning

326 We next assessed whether the same policy could generalize to directional maneuvers. In the turning
 327 task, the model was instructed to walk at a forward velocity of 3 cm/s while executing a leftward
 328 turn at 10 rad/s. As shown in Figure 7, the simulated fly successfully produced a smooth curved
 329 trajectory by modulating stride lengths asymmetrically across the body: legs on the inner side of the
 330 turn reduced their stance amplitude, while contralateral legs extended their strides. This modulation
 331 of gait symmetry arises naturally from the network dynamics, without requiring task-specific tuning
 332 or additional control rules.

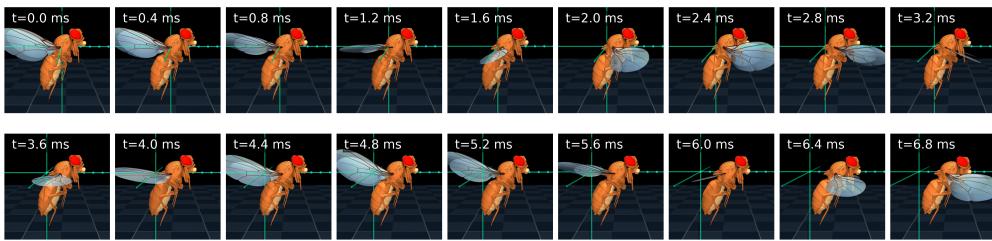


342 **Figure 7: Turning dynamics.** Snapshots of the virtual fly executing a high-speed left turn at 3 cm/s
 343 and 10 rad/s. Over the course of the trajectory, stride lengths on the turning side decrease while
 344 those on the contralateral side increase, producing a smooth curved trajectory. This asymmetry
 345 demonstrates that flyGNN can generalize beyond straight walking to produce directed maneuvers.

346 The ability to perform both straight walking and turning indicates that the learned flyGNN policy
 347 does not simply memorize a single stereotyped gait, but rather encodes a flexible control strategy
 348 that can adapt to new locomotor demands. These findings highlight the robustness of the architecture
 349 and suggest its potential for modeling a wider repertoire of multi-task behaviors.

350
351 **Flight**

352 To assess whether flyGNN can generalize beyond terrestrial locomotion, we additionally trained the
 353 fly to perform a flying task. In this setting, the policy served as a higher-level neural controller that
 354 modulated the output of the wing-beat pattern generator, thereby enabling stable flight dynamics.



364 **Figure 8: Flight dynamics.** Snapshots of the virtual fly executing a straight flight task at a velocity
 365 of 20 cm/s.

367 As shown in Figure 8, the trained controller maintained stable forward flight at a constant speed and
 368 kept body orientation aligned with the target direction, demonstrating that the connectome-based
 369 network can extend from walking to flight locomotion.

370 These results suggest that connectome-structured networks are not limited to walking but can sup-
 371 port a broader repertoire of multimodal behaviors, highlighting the generality of this modeling
 372 framework for embodied control.

373
374
375
376
377

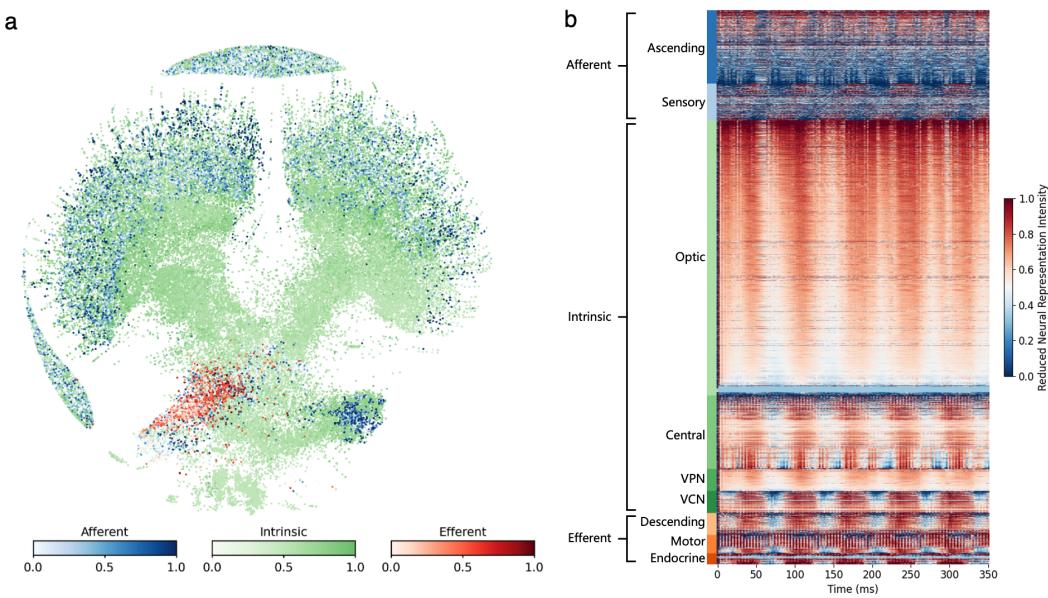
378
379**Neural Representations Analysis**380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399

Figure 9: Neural representations during turning task (0–350 ms). (a) Force-directed graph layout of the fly connectome at 200 ms, with neurons colored by flow annotations (afferent: blue, intrinsic: green, efferent: red) and reduced neural representation intensity. (b) Temporal dynamics of reduced neural representation intensity from 0–350 ms, grouped by flow and superclass annotations. Each row corresponds to one neuron, sorted within its superclass by similarity, and group widths are schematic rather than scaled to neuron counts. The visualization reveals rhythmic patterns aligned with the gait cycle and distinct functional differentiation across superclasses. These results indicate that functional specialization of neurons emerges from the connectome during locomotor control.

One of the key motivations for implementing locomotion through a connectome-based architecture is the opportunity to analyze internal neural representations and study how information propagates through the network. By examining the activity patterns of individual graph nodes, which correspond to neurons, we can ask whether structured and meaningful dynamics emerge, thereby offering a form of connectome-level interpretability. To probe this question, we rolled out trained flyGNN model in simulation and recorded the representations of all neurons at every timestep. These high-dimensional representations were reduced and normalized, yielding the quantity we term reduced neural representation intensity, a proxy for neural activity similar to measures commonly used in neuroscience.

We visualized these reduced neural representations in two complementary ways. First, force-directed graph layouts were used to highlight how intensity distributes across the connectome topology. As shown in Figure 9(a), the snapshot at 200 ms of a turning task reveals structured spatial patterns. When paired with synchronized video of locomotion (available on our anonymous project page), these dynamics reveal that subsets of neurons exhibit rhythmic and heterogeneous oscillations distributed across superclasses. This suggests that the graph’s structure itself induces meaningful partitioning of activity rhythms.

Second, we adopted visualization which aggregates representations by annotated flow and superclass labels. We employed random down-sampling to balance visibility across groups. Within each superclass, neurons were reordered using a spectral method based on the Fiedler vector of the Laplacian constructed from similarity matrices of neural activity, thereby clustering neurons with similar dynamics. Despite the fact that our model only encodes neurons by flow type rather than by finer superclass distinctions, the result shown in Figure 9(b) highlights distinct temporal organization across superclasses. The figure demonstrate that reduced neural representation intensities not only align with locomotor rhythms but also uncover emergent specialization across superclasses, showing how connectomic topology alone drives differentiated functional roles. Similar patterns were also

432 observed in straight-walking tasks (see Appendix E), indicating that these findings are consistent
 433 across episodes and tasks.

434 Taken together, these findings demonstrate that flyGNN develops functional specialization among
 435 neuronal groups directly from connectomic structure, without explicit supervision. The emergence
 436 of distinct, task-aligned activity profiles suggests that the architecture not only generates behav-
 437 iorally valid locomotion but also yields potentially interpretable internal representations.

438

439 5 CONCLUSION AND DISCUSSION

440

441 This study demonstrates that the fly-connectomic Graph Neural Network can serve as a reinforce-
 442 ment learning controller for the high-dimensional dynamical systems of a simulated fruit fly to
 443 achieve diverse movement tasks such as gait initiation, straight walking and turning. By structuring
 444 information flow according to the FlyWire connectome, flyGNN replaces fully connected policy
 445 networks with a graph-based network that directly reflect the wiring diagram of the brain.

446 These results demonstrate that structural priors at the connectome scale provide a powerful inductive
 447 bias for embodied reinforcement learning. Even when simplified to an unweighted directed graph
 448 without synapse counts or neurotransmitter types, the connectome is sufficient to drive diverse high-
 449 dimensional motor control. This suggests that wiring diagrams, long viewed as static anatomical
 450 maps, can be directly instantiated as functional networks for closed-loop control.

451 From the perspective of machine learning, flyGNN offers a new design paradigm for neural ar-
 452 chitecture design. Instead of relying on existing artificial structures, we can adapt structure from
 453 evolved biological networks. This provides a systematic alternative to generic architectures, poten-
 454 tially improving data efficiency, stability, and transferability across tasks. The approach also opens
 455 the possibility of scaling to larger connectomes and more complex agents, where the inductive biases
 456 of real nervous systems may be particularly advantageous.

457 Several limitations highlight opportunities for future progress. First, our implementation simplified
 458 biophysical details, leading to the loss of important biological information. Incorporating richer
 459 information could improve biological fidelity. Second, compared to MLP-based controllers, our
 460 model requires longer per-step computation and higher memory usage, which could be improved.
 461 Finally, extending the framework beyond locomotion will provide a more comprehensive test of the
 462 generality of connectome-based control.

463 In summary, flyGNN demonstrates that whole-brain connectomes can be used for embodied motor
 464 control. By grounding policy architectures in biological wiring diagrams, this approach suggests a
 465 possible towards more human-aligned AI systems, where the inductive biases that shape adaptive
 466 behavior in animals can be systematically transferred to artificial agents.

467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485

486 ETHICS STATEMENT
487488 This work does not involve human or animal subjects, private or sensitive data, or potentially harmful
489 applications. The research uses publicly available connectome datasets and simulated environments,
490 and follows the ICLR Code of Ethics in ensuring fairness, transparency, and responsible stewardship
491 of trustworthy research.492
493 REPRODUCIBILITY STATEMENT
494495 We provide detailed model descriptions and training procedures in the main text, with implementa-
496 tion details in the appendix. Methods for visualizations are described, and source code is available
497 at our anonymous webpage.
498499
500 LARGE LANGUAGE MODELS USAGE STATEMENT
501502 We acknowledge the use of Large Language Models (LLMs) to assist with improving the grammar
503 and clarity of the manuscript. The authors carefully reviewed and verified all content to ensure
504 accuracy and correctness. The authors take full responsibility for the final version of this work.505
506 REFERENCES
507508 Azevedo, A. et al. Connectomic reconstruction of a female *drosophila* ventral nerve cord.
509 *Nature*, 631(8020):360–368, July 2024. ISSN 0028-0836, 1476-4687. doi: 10.1038/
510 s41586-024-07389-x.511 Cheng, X. et al. Expressive whole-body control for humanoid robots, 2024. URL <https://arxiv.org/abs/2402.16796>.512 Corso, G., Cavalleri, L., Beaini, D., Liò, P. and Veličković, P. Principal neighbourhood aggregation
513 for graph nets, December 2020.514 Ding, Y., Pandala, A., Li, C., Shin, Y.H. and Park, H.W. Representation-free model predictive control
515 for dynamic motions in quadrupeds. *IEEE Transactions on Robotics*, 37(4):1154–1171, August
516 2021. ISSN 1941-0468. doi: 10.1109/tro.2020.3046415. URL <http://dx.doi.org/10.1109/TRO.2020.3046415>.517 Dorkenwald, S. et al. Neuronal wiring diagram of an adult brain. *Nature*, 634(8032):124–138,
518 October 2024. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-024-07558-y.519 Fey, M. and Lenssen, J.E. Fast graph representation learning with PyTorch Geometric. In *ICLR*
520 *Workshop on Representation Learning on Graphs and Manifolds*, 2019.521 Fey, M. et al. PyG 2.0: Scalable learning on real world graphs. In *Temporal Graph Learning*
522 *Workshop at KDD*, 2025.523 Hamilton, W.L., Ying, R. and Leskovec, J. Inductive representation learning on large graphs,
524 September 2018.525 He, K., Zuo, C., Ma, C. and Sui, Y. Dynsyn: Dynamical synergistic representation for efficient
526 learning and control in overactuated embodied systems, 2024. URL <https://arxiv.org/abs/2407.11472>.527 Kipf, T.N. and Welling, M. Semi-supervised classification with graph convolutional networks,
528 February 2017.

529 Kobourov, S.G. Spring embedders and force directed graph drawing algorithms, January 2012.

530 Kumar, A., Fu, Z., Pathak, D. and Malik, J. Rma: Rapid motor adaptation for legged robots, 2021.
531 URL <https://arxiv.org/abs/2107.04034>.

540 Lappalainen, J.K. et al. Connectome-constrained networks predict neural activity across the
 541 fly visual system. *Nature*, 634(8036):1132–1140, 2024. ISSN 0028-0836. doi: 10.1038/s41586-024-07939-3.
 542

543 Lesser, E. et al. Synaptic architecture of leg and wing premotor control networks in drosophila.
 544 *Nature*, 631(8020):369–377, July 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07600-z.
 545

546 Lobato-Rios, V. et al. NeuroMechFly, a neuromechanical model of adult *Drosophila melanogaster*.
 547 *Nature Methods*, 19(5):620–627, May 2022. doi: 10.1038/s41592-022-01466-7. URL <https://doi.org/10.1038/s41592-022-01466-7>.
 548

549 McInnes, L., Healy, J., Saul, N. and Grossberger, L. Umap: Uniform manifold approximation and
 550 projection. *The Journal of Open Source Software*, 3(29):861, 2018.
 551

552 Schlegel, P. et al. Whole-brain annotation and multi-connectome cell typing of drosophila.
 553 *Nature*, 634(8032):139–152, October 2024. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-024-07686-5.
 554

555 Shiu, P.K. et al. A drosophila computational brain model reveals sensorimotor processing. *Nature*,
 556 634(8032):210–219, October 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07763-9.
 557

558 Todorov, E., Erez, T. and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012
 559 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE,
 560 2012. doi: 10.1109/IROS.2012.6386109.

561 Vaxenburg, R. et al. Whole-body physics simulation of fruit fly locomotion. *Nature*, 643(8074):
 562 1312–1320, July 2025. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-025-09029-4.
 563

564 Veličković, P. et al. Graph attention networks, February 2018.

565 Wang, Y. et al. Dynamic graph cnn for learning on point clouds, June 2019.

566

567 Wang-Chen, S. et al. Neuromechfly v2: simulating embodied sensorimotor control in
 568 adult drosophila. *Nature Methods*, 21(12):2353–2362, November 2024. ISSN 1548-
 569 7105. doi: 10.1038/s41592-024-02497-y. URL <http://dx.doi.org/10.1038/s41592-024-02497-y>.
 570

571 Wei, Y., Zhuang, S., Zhuang, V. and Sui, Y. Motion control of high-dimensional musculoskeletal
 572 systems with hierarchical model-based planning, 2025. URL <https://arxiv.org/abs/2505.08238>.
 573

574

575 Zheng, Z. et al. A complete electron microscopy volume of the brain of adult drosophila
 576 *melanogaster*. *Cell*, 174(3):730–743.e22, July 2018. ISSN 00928674. doi: 10.1016/j.cell.2018.
 577 06.019.

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594 **A DETAILS OF MODEL ARCHITECTURE AND TRAINING PROCESS**
595596 **A.1 MODEL ARCHITECTURE AND IMPLEMENTATION**
597598 The flyGNN model is implemented using the PyTorch Geometric (PyG) (Fey & Lenssen, 2019; Fey
599 et al., 2025) library to construct a graph neural network that directly mirrors the Drosophila connec-
600 tome. The network architecture is structured as a directed graph where nodes represent neurons and
601 edges represent synaptic connections, partitioned into afferent, intrinsic, and efferent sets based on
602 information flow. The model processes inputs through the following components:603 Input Normalization: A RunningNorm layer is applied to observations to stabilize training by main-
604 taining running estimates of mean and variance. This layer operates online and is updated during
605 training.606 Encoder: Observations are projected into the afferent neuron state space via a linear layer followed
607 by ReLU activation.608 Graph Propagation: Information is propagated through the connectome using a message-passing
609 mechanism with gated updates combining previous states and incoming messages.610 Decoder: Efferent neuron states are aggregated and passed through a multi-layer perceptron with
611 ReLU activations, outputting action means and standard deviations via linear and softplus heads,
612 respectively.613 The model uses simplified biological assumptions—connections are treated as unweighted and di-
614 rected, ignoring synaptic counts or neurotransmitter types. All experiments were conducted on
615 servers equipped with NVIDIA A100 80GB PCIe GPUs and Intel Xeon Gold 6348 CPUs.616 **A.2 TRAINING PIPELINE**
617

618 Training proceeds in two stages:

619 Imitation Learning: The policy is initialized by mimicking expert trajectories generated by a pre-
620 trained MLP controller provided by flybody. This stage uses PyTorch Lightning for distributed
621 training across multiple workers.622 Reinforcement Learning: The model is fine-tuned with Proximal Policy Optimization to maximize
623 task rewards. This stage is implemented using PyTorch Distributed Data Parallel (DDP) for scal-
624 ability.625 Optimization uses AdamW with a learning rate scheduler (ReduceLROnPlateau) that reduces the
626 learning rate upon validation loss improving. Both stages employ gradient clipping to ensure stabil-
627 ity.628 **A.3 TASK-SPECIFIC CONFIGURATIONS**
629

630 Walking Task:

631

- 632 • Observation dimension: 1,253 (proprioceptive + exteroceptive + visual inputs),
- 633 • Action dimension: 59 (joint actuators and adhesion controls),
- 634 • Learning rate: 1×10^{-4} ,
- 635 • Message-Passing: GraphSAGE, 16 node channels, 4 message-passing layers,
- 636 • Expert policy architecture: 512–512–512–512 fully-connected layers with LayerNorm +
- 637 Tanh (first layer) and ELU activations, outputting mean (Linear) and std (Linear + Softplus)
- 638 for 59-dimensional actions.

639 Flight Task:

640

- 641 • Observation dimension: 104 (simplified proprioceptive and kinematic inputs),
- 642 • Action dimension: 12 (wing torques, pattern generator modulation, body joints),
- 643 • Learning rate: 1×10^{-5} ,

- Message-Passing: GraphSAGE, 8 node channels, 4 message-passing layers,
- Expert policy architecture: 256–256–256 fully-connected layers with LayerNorm + Tanh (first layer) and ELU activations, outputting mean (Linear) and std (Linear + Softplus) for 12-dimensional actions.

Both tasks use identical graph topology derived from the FlyWire connectome but differ in input/output dimensions and network hyperparameters due to behavioral constraints. Training metrics are logged for performance monitoring.

B DETAILED SETTINGS OF LOCOMOTION ENVIRONMENT

Our experiments follow the locomotor task design introduced in the flybody simulator, which provides imitation-learning datasets for terrestrial (*walking*) and aerial (*flight*) behaviors.

We evaluate four tasks:

- Gait initiation: generating stable stepping patterns from rest,
- Straight-line walking: tracking forward centre-of-mass (CoM) trajectories,
- Turning: executing lateral turns at constant speed,
- Flight: stabilizing and steering free flight trajectories.

For walking-based tasks, the default setting of flybody receives a 741-dimensional proprioceptive/exteroceptive observation, comprising:

- accelerometer (3), gyro (3), velocimeter (3), world z -axis (3),
- actuator activations (59),
- appendage pose (21), force sensors (18),
- joint positions (85) and velocities (85),
- tactile contacts (6),
- reference displacement (195) and root quaternion (260).

We augment this with binocular visual input: left and right eye cameras ($32 \times 32 \times 3$ RGB each) downsampled and resized to two 16×16 grayscale figures. The final observation dimension is 1,253.

Actions remain 59-dimensional, actuating adhesion, head/abdomen motion, and all leg joints as in default settings of flybody walking task.

For the flight task, we keep the flybody sensory design with a 104-dimensional input comprising:

- accelerometer (3), gyro (3), velocimeter (3), world z -axis (3),
- joint positions (25) and velocities (25),
- reference displacement (18) and root quaternion (24).

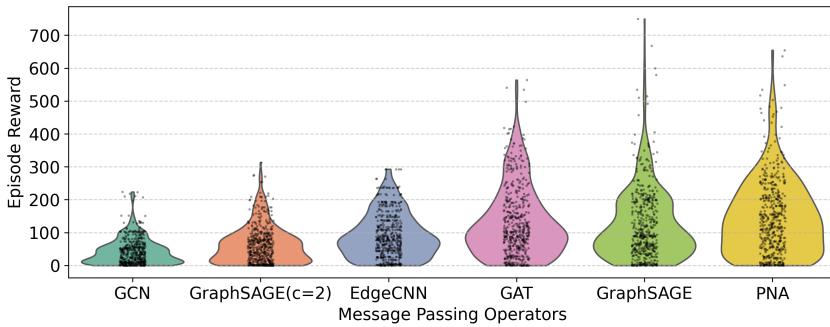
The policy outputs 12 control signals: instantaneous wing torques, head/abdomen angles, and Wing-Pattern Generator (WPG) frequency modulation. The WPG provides a nominal wing-beat template, while the policy learns residual corrections.

C ABLATION STUDIES ON MESSAGE-PASSING OPERATORS

We evaluated flyGNN across several message-passing operators, including GCN (Kipf & Welling, 2017), EdgeCNN (Wang et al., 2019), GAT (Veličković et al., 2018), GraphSAGE (Hamilton et al., 2018) and PNA (Corso et al., 2020). Table 1 shows the evaluation KL loss and the average reward within 500 episodes of different message-passing operators. We used the imitation learning dataset provided by flybody, and because different imitation trajectories exhibit variability, the rewards naturally show high fluctuation. The results show that simpler operators (e.g., GCN, EdgeCNN) generally converged with higher KL divergence to the teacher and lower average reward, while more

702
703 Table 1: Performance comparison of flyGNN with different message-passing operators.
704

Model	Node dim	Depth	Eval KL ↓	Avg. reward ↑
GCN	16	4	7.93	43.13
GraphSAGE($c = 2$)	2	6	6.85	62.85
EdgeCNN	4	2	5.54	90.48
GAT	8	2	3.43	132.65
GraphSAGE	16	4	3.31	125.55
PNA	4	2	2.89	145.33

725
726 Figure 10: **Performance across message-passing operators.** Distribution of episode rewards ob-
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
10010
10011
10012
10013
10014
10015
10016
10017
10018
10019
10020
10021
10022
10023
10024
10025
10026
10027
10028
10029
10030
10031
10032
10033
10034
10035
10036
10037
10038
10039
10040
10041
10042
10043
10044
10045
10046
10047
10048
10049
10050
10051
10052
10053
10054
10055
10056
10057
10058
10059
10060
10061
10062
10063
10064
10065
10066
10067
10068
10069
10070
10071
10072
10073
10074
10075
10076
10077
10078
10079
10080
10081
10082
10083
10084
10085
10086
10087
10088
10089
10090
10091
10092
10093
10094
10095
10096
10097
10098
10099
100100
100101
100102
100103
100104
100105
100106
100107
100108
100109
100110
100111
100112
100113
100114
100115
100116
100117
100118
100119
100120
100121
100122
100123
100124
100125
100126
100127
100128
100129
100130
100131
100132
100133
100134
100135
100136
100137
100138
100139
100140
100141
100142
100143
100144
100145
100146
100147
100148
100149
100150
100151
100152
100153
100154
100155
100156
100157
100158
100159
100160
100161
100162
100163
100164
100165
100166
100167
100168
100169
100170
100171
100172
100173
100174
100175
100176
100177
100178
100179
100180
100181
100182
100183
100184
100185
100186
100187
100188
100189
100190
100191
100192
100193
100194
100195
100196
100197
100198
100199
100200
100201
100202
100203
100204
100205
100206
100207
100208
100209
100210
100211
100212
100213
100214
100215
100216
100217
100218
100219
100220
100221
100222
100223
100224
100225
100226
100227
100228
100229
100230
100231
100232
100233
100234
100235
100236
100237
100238
100239
100240
100241
100242
100243
100244
100245
100246
100247
100248
100249
100250
100251
100252
100253
100254
100255
100256
100257
100258
100259
100260
100261
100262
100263
100264
100265
100266
100267
100268
100269
100270
100271
100272
100273
100274
100275
100276
100277
100278
100279
100280
100281
100282
100283
100284
100285
100286
100287
100288
100289
100290
100291
100292
100293
100294
100295
100296
100297
100298
100299
100300
100301
100302
100303
100304
100305
100306
100307
100308
100309
100310
100311
100312
100313
100314
100315
100316
100317
100318
100319
100320
100321
100322
100323
100324
100325
100326
100327
100328
100329
100330
100331
100332
100333
100334
100335
100336
100337
100338
100339
100340
100341
100342
100343
100344
100345
100346
100347
100348
100349
100350
100351
100352
100353
100354
100355
100356
100357
100358
100359
100360
100361
100362
100363
100364
100365
100366
100367
100368
100369
100370
100371
100372
100373
100374
100375
100376
100377
100378
100379
100380
100381
100382
100383
100384
100385
100386
100387
100388
100389
100390
100391
100392
100393
100394
100395
100396
100397
100398
100399
100400
100401
100402
100403
100404
100405
100406
100407
100408
100409
100410
100411
100412
100413
100414
100415
100416
100417
100418
100419
100420
100421
100422
100423
100424
100425
100426
100427
100428
100429
100430
100431
100432
100433
100434
100435
100436
100437
100438
100439
100440
100441
100442
100443
100444
100445
100446
100447
100448
100449
100450
100451
100452
100453
100454
100455
100456
100457
100458
100459
100460
100461
100462
100463
100464
100465
100466
100467
100468
100469
100470
100471
100472
100473
100474
100475
100476
100477
100478
100479
100480
100481
100482
100483
100484
100485
100486
100487
100488
100489
100490
100491
100492
100493
100494
100495
100496
100497
100498
100499
100500
100501
100502
100503
100504
100505
100506
100507
100508
100509
100510
100511
100512
100513
100514
100515
100516
100517
100518
100519
100520
100521
100522
100523
100524
100525
100526
100527
100528
100529
100530
100531
100532
100533
100534
100535
100536
100537
100538
100539
100540
100541
100542
100543
100544
100545
100546
100547
100548
100549
100550
100551
100552
100553
100554
100555
100556
100557
100558
100559
100560
100561
100562
100563
100564
100565
100566
100567
100568
100569
100570
100571
100572
100573
100574
100575
100576
100577
100578
100579
100580
100581
100582
100583
100584
100585
100586
100587
100588
100589
100590
100591
100592
100593
100594
100595
100596
100597
100598
100599
100600
100601
100602
100603
100604
100605
100606
100607
100608
100609
100610
100611
100612
100613
100614
100615
100616
100617
100618
100619
100620
100621
100622
100623
100624
100625
100626
100627
100628
100629
100630
100631
100632
100633
100634
100635
100636
100637
100638
100639
100640
100641
100642
100643
100644
100645
100646
100647
100648
100649
100650
100651
100652
100653
100654
100655
100656
100657
100658
100659
100660
100661
100662
100663
100664
100665
100666
100667
100668
100669
100670
100671
100672
100673
100674
100675
100676
100677
100678
100679
100680
100681
100682
100683
100684
100685
100686
100687
100688
100689
100690
100691
100692
100693
100694
100695
100696
100697
100698
100699
100700
100701
100702
100703
100704
100705
100706
100707
100708
100709
100710
100711
100712
100713
100714
100715
100716
100717
100718
100719
100720
100721
100722
100723
100724
100725
100726
100727
100728
100729
100730
100731
100732
100733
100734
100735
100736
100737
100738
100739
100740
100741
100742
100743
100744
100745
100746
100747
100748
100749
100750
100751
100752
100753
100754
100755
100756
100757
100758
100759
100760
100761
100762
100763
100764
100765
100766
100767
100768
100769
100770
100771
100772
100773
100774
100775
100776
100777
100778
100779
100780
100781
100782
100783
100784
100785
100786
100787
100788
100789
100790
100791
100792
100793
100794
100795
100796
100797
100798
100799
100800
100801
100802
100803
100804
100805
100806
100807
100808
100809
100810
100811
100812
100813
100814
100815
100816
100817
100818
100819
100820
100821
100822
100823
100824
100825
100826
100827
100828
100829
100830
100831
100832
100833
100834
100835
100836
100837
100838
100839
100840
100841
100842
100843
100844
100845
100846
100847
100848
100849
100850
100851
100852
100853
100854
100855
100856
100857
100858
100859
100860
100861
100862
100863
100864
100865
100866
100867
100868
100869
100870
100871
100872
100873
100874
100875
100876
100877
100878
100879
100880
100881
100882
100883
100884
100885
100886
100887
100888
100889
100890
100891
100892
100893
100894
100895
100896
100897
100898
100899
100900
100901
100902
100903
100904
100905
100906
100907
100908
100909
100910
100911
100912
100913
100914
100915
100916
100917
100918
100919
100920
100921
100922
100923
100924
100925
100926
100927
100928
100929
100930
100931
100932
100933
100934
100935
100936
100937
100938
100939
100940
100941
100942
100943
100944
100945
100946
100947
100948
100949
100950
100951
100952
100953
100954
100955
100956
100957
100958
100959
100960
100961
100962
100963
100964
100965
100966
100967
100968
100969
100970
100971
100972
100973
100974
100975
100976
100977
100978
100979
100980
100981
100982
100983
100984
100985
100986
100987
100988
100989
100990
100991
100992
100993
100994
100995
100996
100997
100998
100999
1001000
1001001
1001002
1001003
1001004
1001005
1001006
1001007
1001008
1001009
1001010
1001011
1001012
1001013
1001014
1001015
1001016
1001017
1001018
1001019
1001020
1001021
1001022
1001023
1001024
1001025
1001026
1001027
1001028
1001029
1001030
1001031
1001032
1001033
1001034
1001035
1001036
1001037
1001038
1001039
1001040
1001041
1001042
1001043
1001044
1001045
1001046
1001047
1001048
1001049
1001050
1001051
1001052
1001053
1001054
1001055
1001056
1001057
1001058
1001059
1001060
1001061
1001062
1001063
1001064
1001065
1001066
1001067
1001068
1001069
1001070
1001071
1001072
1001073
1001074
1001075
1001076
1001077
1001078
1001079
1001080
1001081
1001082
1001083
1001084
1001085
1001086
1001087
1001088
1001089
1001090
1001091
1001092
1001093
1001094
1001095
1001096
1001097
1001098
1001099
1001100
1001101
1001102
1001103
1001104
1001105
1001106
1001107
1001108
1001109
1001110
1001111
1001112
1001113
1001114
1001115
1001116
1001117
1001118
1001119
1001120
1001121
1001122
1001123
1001124
1001125
1001126
1001127
1001128
100112

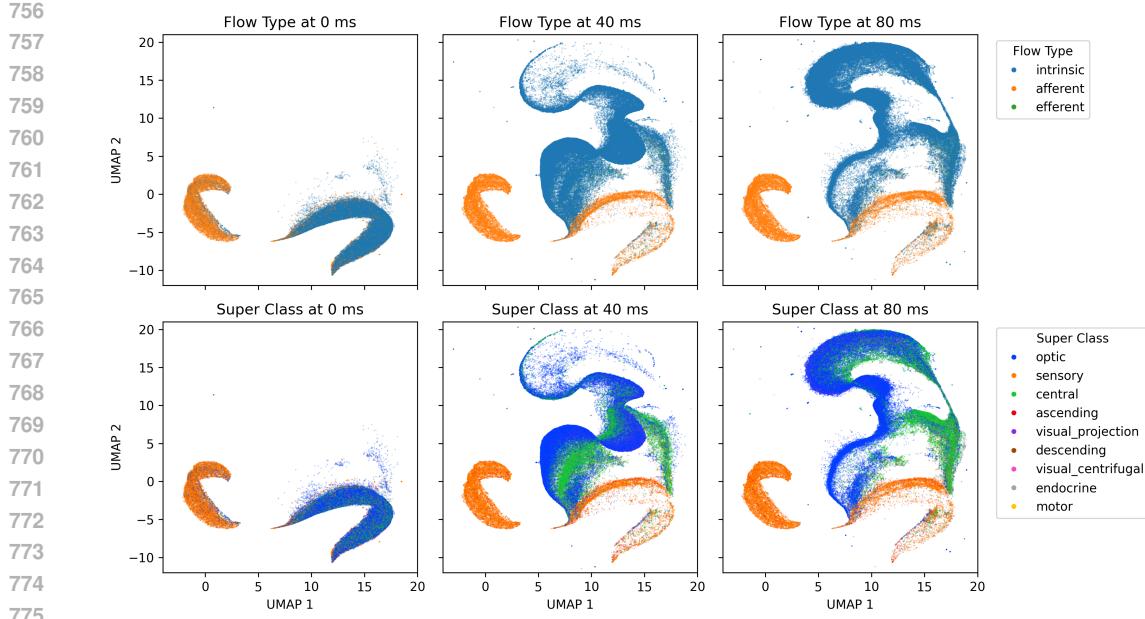


Figure 11: **Neuron state representation during gait initiation.** UMAP projections of neural features at different time points (0 ms, 40 ms, 80 ms). Top: neurons colored by FlyWire flow type (afferent, intrinsic, efferent). Bottom: neurons colored by ‘superclass’ annotations. The visualization shows that initially mixed features gradually separate into distinct clusters, with optic and central populations becoming clearly differentiated by 80 ms. This structured reorganization indicates that the flyGNN policy induces meaningful neural dynamics, supporting functional specialization as gait coordination emerges.

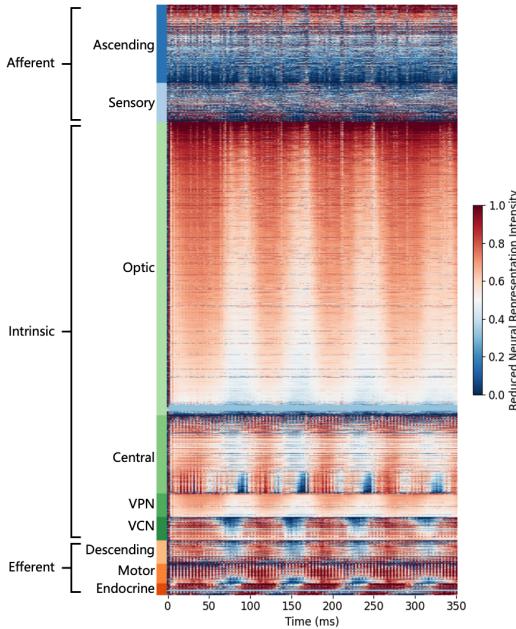


Figure 12: **Neural representations during straight walking (0–350 ms).** Temporal dynamics of reduced neural representations, grouped by flow and superclass annotations. Widths of superclass groups are schematic and not proportional to actual neuron counts. The visualization reveals rhythmic activity aligned with gait cycles, while motor and descending groups show sparser but phase-locked dynamics. Unlike turning, the straight-walking condition lacks the optic phase shift, indicating more symmetric activation patterns during forward locomotion.

810 E VISUALIZATION OF NEURAL REPRESENTATIONS IN WALKING TASK
811812 To complement the turning-task analysis, we also visualized reduced neural representation intensities
813 during walking task (see Sec. 4). Figure 12 shows temporal dynamics of neural activity from
814 0–350 ms, including the first 80 ms of gait initiation, after dimensionality reduction with principal
815 component analysis and normalization. Neurons are grouped by flow type and superclass annotations,
816 with width adjusted schematically for visibility, and reordered within each superclass using
817 spectral similarity sorting.818 The visualization highlights distinct temporal activation profiles across neural groups. In particular,
819 optic and central populations display rhythmic modulations aligned with gait cycles, while motor
820 and descending neurons show sparser but phase-locked activation. Compared to the turning condition,
821 the straight-walking trajectories exhibit more symmetric activation patterns without the optic
822 phase shift observed during turning. This difference suggests that lateralized activation dynamics
823 may specifically support asymmetric motor outputs during directional maneuvers, whereas straight
824 walking is driven by more synchronized, globally coordinated patterns.825
826 F DETAILED METHOD FOR VISUALIZATION
827828 F.1 FIGURE 2(A)
829830 We visualized the aggregated synapse graph of the fly connectome using the FlyWire FAFB v783
831 dataset, with nodes classified by superclass labels provided by FlyWire. The node size reflects the
832 number of neurons in each class, while the thickness and darkness of the directed edges represent
833 the number of aggregated synaptic connections.834 F.2 FIGURE 2(B)
835836 In the visualization of the network of connectome, we did not incorporate the three-dimensional
837 structural priors of *Drosophila* neurons. Instead, we employed a force-directed layout algo-
838 rithm (Kobourov, 2012) for graph drawing. In this method, edges are modeled as springs while
839 nodes exert repulsive forces on each other, and the system iteratively evolves toward a low-energy
840 equilibrium. This layout highlights the topological organization of the network, making the rela-
841 tionships among different super-classes and subgroups more visually apparent. For visualization,
842 we applied a threshold of more than 25 synapses and retained only connections exceeding this cut-
843 off.844 F.3 FIGURE 9(B)
845846 Figure 9(b) presents the temporal dynamics of reduced neural representation intensity across the
847 fly connectome during a 350 ms turning maneuver. To visualize population activity patterns, we
848 first extracted high-dimensional neural features from the model, with shape $T \times N \times C$ (time \times
849 neurons \times channels). These features were compressed into a single-channel intensity measure via
850 PCA, retaining only the first principal component along the channel dimension. The resulting values
851 were then clipped to the 5th–95th percentile range and min-max normalized to $[0, 1]$ to improve
852 comparability and reduce outlier effects.853 Due to the extreme imbalance in superclass sizes (e.g., > 77000 neurons in optic classes versus
854 only 106 in motor classes), we employed stratified random downsampling to ensure visibility across
855 all functional groups while preserving relative proportions. The downsampling thresholds were
856 set as follows: sensory: 400, ascending: 200, optic: 1500, central: 400, visual projection: 120,
857 visual centrifugal: 120, descending: 120, motor: 100, endocrine: 60. This approach maintained the
858 diversity of neural responses while creating visually interpretable group representations.859
860
861
862
863