Computational Proxies for Dopamine-Mediated Learning in the Olfactory System of Mosquitoes

Suh Woo Jung

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Electrical Engineering

University of Washington

2019

Reading Committee:

Eli Shlizerman

Jeff Riffell

Sam A. Burden

Program Authorized to Offer Degree:

Electrical and Computer Engineering

University of Washington

Abstract

Computational Proxies for Dopamine-Mediated Learning in the Olfactory System of Mosquitoes

Suh Woo Jung

Chair of the Supervisory Committee:
Eli Shlizerman
Electrical and Computer Engineering

Olfactory learning in mosquitoes plays an essential role in determining their host preference and has significant epidemiological consequences. Dopamine effects on olfactory learning in mosquito brain are still unclear, although it is known to actively modulate the neural activity of Projection Neurons (PNs) in the antennal lobe (AL). To investigate this, we study the firing patterns of the PNs before, during, and after the superfusion of dopamine over the brain of the mosquito. In particular, we compute the fixed points in classifying odor space corresponding to each firing pattern. Comparison of the locations of fixed points in each phase of the experiment indicates that the superfusion of dopamine causes fixed points to dislocate. To investigate how neurons facilitate such function in the AL, we extend a lateral inhibition firing-rate model of the AL to include dopamine neurons. We then implement various time-dependent learning rules

(gradient based, Hebbian learning, reward-based) that minimize the norm difference between expected firing rates and observed ones to quantify the connectome weights of the Dopaminergic (DA) neurons. The comparison of the predicted trajectories and updated connection matrices reveal that these update methods have different element-to-element modulation yet the similar pattern of modulation during and after the application of dopamine. Furthermore, the successful generation of the firing patterns of the PNs and the relative weights of the lateral inhibition neural connectomes suggest that the DA neurons selectively modulate the neural responses in the AL and follow a selective recovery pattern.

Table of Contents:

1. Introduction

Various studies conducted over the past decade provide evidence which demonstrates the importance of biogenic amines like dopamine and octopamine in learning and memory (Chilaka, Perkins, and Tripet 2012). For dopamine, classic studies suggest that the dopaminergic neurons mediate learning by responding to unpredicted rewards and reward-predicting signals (Schultz 1997). When there is an error in the prediction, the reinforcer signals induce learning in neural dynamics (Berke 2018). This mechanism inspired many reinforcement learning methods, including the temporal difference (TD) algorithm and the Critic-Actor architecture (Suri and Schultz 1999; Joel, Niv, and Ruppin 2002).

Superfusing dopamine on the antennal lobe (AL), the olfactory center of the brain, is known to strongly modulate the activities of neurons and increase the ability to learn new hosts (Vinauger et al. 2018). In our research, we process the multi-neuronal recordings of the projection neurons (PNs) in the AL under three different phases: before (P1), during (P2), and after (P3) the superfusion of dopamine. The firing-rates of PNs were recorded for six different odorants for several trials in each of the phases. The underlying concept of this study is that the transient dynamics of neurons in the mosquito AL will converge to a fixed point given an input under different phases of the experiment (Blaszka et al. 2017). Investigating the neuronal representation of the single odorant is crucial for understanding how the mosquitoes recognize more complex odors and determine their host preferences (Riffell et al. 2014).

To study how dopamine modulation could rearrange neural connectivity in the AL, we use a network model which infers the connectome of neurons in the AL from recorded data. In particular, we employ a lateral inhibition model that is known to facilitate odor discrimination and enhance the contrast between different stimuli (Hirsch and Gilbert 1991; Arevian, Kapoor, and Urban 2008). Using this model, we can characterize the activities of Receptor Neurons (RNs), local interneurons (LNs) and projection neurons (PNs) using ordinary differential equations (Shlizerman, Riffell, and Kutz 2014).

Our computational approach takes the following steps: first, we calibrate the network based on the observed firing rates and predefine the connectivity matrices that accurately depict the trajectories of P1. Then, we further extend the neural dynamics model and propose a new learning rule to accommodate the roles of the Dopaminergic Neurons (DNs) in P2. Finally, we examine the validity of the various learning rules by comparing the results in P3.

We use different training methods including Hebbian learning which is inspired by the dopamine effect on synaptic plasticity and demonstrates high efficiency in predicting the firing rates in the neuronal network (Montague, Dayan, and Sejnowski 1996). Moreover, it has been shown that reward-based Hebbian learning successfully generates complex trajectories in feedforward and recurrent neural networks (Soltoggio and Steil 2013; Hoerzer, Legenstein, and Maass 2014). By comparing the resulting connectivity weights from non-Hebbian and Hebbian Learning, we discuss the efficiency and plausibility of such methods in olfactory learning.

2. Results

2.1 Projection Neurons and Odor Classification Space

2.1.1 Preliminary Analysis of Multi-neuron Recordings

The dataset includes the recordings of the spiking activity from multiple projection neurons (K = 43) in the antennal lobe of mosquitoes exposed to six different odor molecules (odorants): octenol (S1), lactic acid (S2), myrcene (S3), benzaldehyde (S4), ammonia (S5) and ethanol (S6) as shown in **Table 1**. Each response was recorded for 1 second and smoothed with a Gaussian filter. Each odor molecule was presented four times with a 2-minute interval between each trial.

As a preliminary analysis, we compared the firing rates of each neuron in P1 and P2. The firing rates were averaged over all odorants during a realistic response time of 400ms to 600ms. As a result, 51.9% of the neural units showed increased firing rates during P2, 37.2 % showed decreased responses, and the remaining 10.9% demonstrated a neutral effect. Similarly, in P3, 29.1% of the recorded neurons displayed increased firing rates, 58.9% decreased firing rates, and the remaining 12.0% showed no change.

On a single neural level, no distinct pattern was observed from these recorded neurons, as many of them broadly responded to different odors while some neurons were tuned to a single odor. Similarly, dopamine application elicited stronger responses in a selected set of neurons but did not demonstrate global effects. This suggests that dopamine selectively affects the AL encoding space and that a more complex network structure mediates the responses of each projection neuron.

Table 1. Benchmark dataset of multi-neuronal recordings and stimuli

Stimulus Name	Stimulus Label
Octenol (oct)	S1
Lactic acid (la)	S2
Myrcene (myr)	S3
Benzaldehyde (benz)	S4
Ammonia (ammonia)	S5
Ethanol (etoh)	S6

2.1.2 Definition of Library Matrix

First, we construct a library matrix L that stores the observed firing rates of each neuron in response to different odorants in three phases: before (P1), during (P2), and after (P3) dopamine superfusion. The library matrix during P1 is used as a benchmark dataset representing the normal physiological state of the antennal lobe. During P2, dopamine was applied to the brain of the mosquitoes. Neural activity in P3 was recorded after the brains were washed multiple times the next day. Thus, the library matrix used for the analysis consists of three matrices with the dimension of K (number of recorded neurons) times S (number of tested odor stimuli):

$$L_{P1} = \{y_{S1}^{P1}, \dots, y_{Sk}^{P1}\}, L_{P2} = \{y_1^{P2}, \dots, y_s^{P2}\}, L_{P3} = \{y_1^{P3}, \dots, y_s^{P3}\} \quad (1)$$

Here, each column vector, y, represents the activity of the neural population in response to odor S_k. These activities correspond to the averaged activity during the first 600ms after the onset of stimuli.

We then use library matrices in three phases to analyze how each projection neuron in the network increased or decreased activity in the next phase. Let D be the modulation matrix that quantifies the changes in L between two consecutive phases i.e., D_{P2-P1} measured the responses of each neuron during P2 compared to P1. By assigning a modulation matrix that generates the library matrices in the following two phases, and running the convex optimization that minimizes the cost in Eq. (2), we compute the degree of modulation in each neuron.

$$c_1 = \|\bar{L}_{P2} - D_{P2-P1}\,\bar{L}_{P1}\|_2, c_2 = \|\bar{L}_{P3} - D_{P3-P2}\,\bar{L}_{P2}\|_2 \quad (2)$$

$\bar{\mathrm{L}}_{\mathrm{P2}}$ and $\bar{\mathrm{L}}_{\mathrm{P3}}$ represent the library matrices in each phase, where the firing rates of each projection neuron were averaged over trials. The matrix D is initialized as an identity matrix. We can interpret

that neurons with computed D weight > 1 received higher input from neighboring neurons as a result of dopamine modulation and vice versa.

Convex optimization on the control experiment (no stimulus is given) in P2 shows that some neurons generate stronger responses regardless of stimulus, while in P3 neurons are generally inhibited (**Figure 1** Left). PN response to odorants varies from the response to control. For example, the response to lactic acid displays strong modulation during P2 (**Figure 1** Right). Similar to the control experiment, PNs are generally inhibited during P3 even when the odor was present.

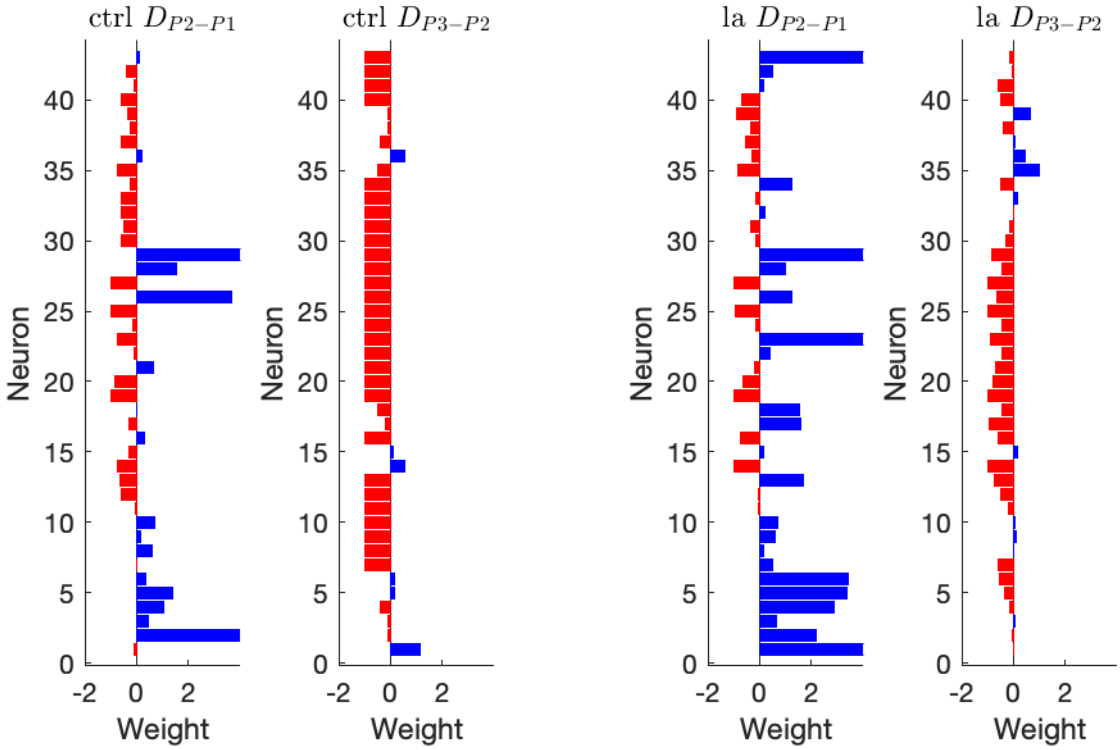

Figure 1. Changes in PN activity from P1 to P2 and P2 to P3 in control experiment (left) and response to lactic acid (right). Neurons with modulation weight higher than 1 (blue) generate stronger response and neurons with weights less than 1 (red) generate weaker than in the previous phase.

2.1.3 Encoding Space and Fixed Points

Next, we apply the Exclusive Threshold Reduction (ETR) method (Blaszka et al. 2017) to PN responses of odorants comprising octanol, lactic acid, myrcene, benzaldehyde, and ammonia as follows. From the library matrix L defined above, we select the maximal value in each row to generate an orthogonal matrix O. The coordinates of an odor fixed point are defined by $L^T O$ and projected onto the low dimensional odor space.

For each odor, we project the fixed points in the encoding space constructed by the orthogonal matrix O, which remains fixed from P1. Essentially, we compare the locations of the fixed points in three phases to investigate how dopamine dislocates them in the initial odor space.

The relocation distances of the fixed point in the pre-dopamine (P1), dopamine (P2) and the post-dopamine phase (P3) in **Figure 2** illustrate that the degree of dopamine modulation in each phase differs from one odor to another. For example, ammonia fixed point remains nearly intact in all phases, whereas EtOH fixed points are displaced farther as in P2 and P3.

The summary of calculated Euclidean distances for all odors is presented in **Table 2**. For all odors, the fixed points are most separated in P3, which indicates that activities of PNs hardly recover to the original state. The most significant displacement is observed for ethanol with the Euclidean distance of 11.14 from P1 to P3. The fixed point of myrcene is displayed the least from P1 to P2 (1.54).

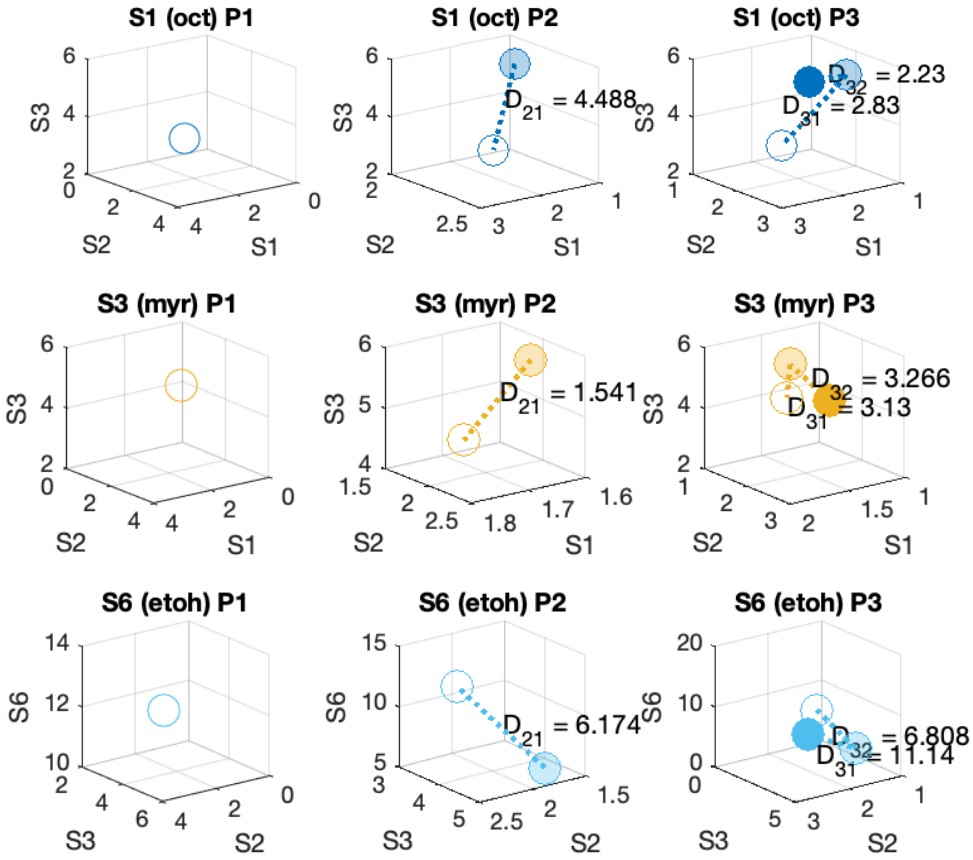

Figure 2. Fixed points of selected odor stimulus (octenol, benzaldehyde, and ethanol) in three phases. D_{ij} indicates the Euclidean distance from phase j to i in the odor space constructed by O in P1.

Table 2. Fixed point relocation distances of each odor stimuli.

Stimulus Name & Label	Norm distance from P1 to P2	Norm distance from P2 to P3	Norm distance from P1 to P3
Octenol (S1)	4.487	2.229	2.829
Lactic Acid (S2)	3.403	3.725	2.621
Myrcene (S3)	1.541	3.266	3.133
Benzaldehyde (S4)	4.373	3.773	5.769
Ammonia (S5)	3.169	2.426	4.624
Ethanol (S6)	6.174	6.807	11.13

2.2 Lateral Inhibition Model

2.2.1 Model Definition and System of Equations

We employ lateral inhibition where both inhibitory and excitatory neurons receive common input and interact to mediate the response of the PNs (**Figure 3**). To analyze dopamine mediation of the connectivity of the AL network, we adopt a model that characterizes the lateral-inhibitory neurons and the excitatory neurons in the AL:

$$\frac{dx}{dt} = -x + J \tag{3}$$

$$\frac{dy}{dt} = -\beta y + [Ax - Bz]^+ \tag{4}$$

$$\frac{dz}{dt} = -\gamma z + [Cx - Ez]^+ \tag{5}$$

The vectors x, y, and z represent the activities of the RNs, LNs and the PNs respectively. The network connectomes, A, B, C and E, each enhances or inhibits inputs from the RNs and the LNs. The inputs are modulated by a standard linear threshold function denoted by $[.]^+$ (Shlizerman, Riffell, and Kutz 2014).

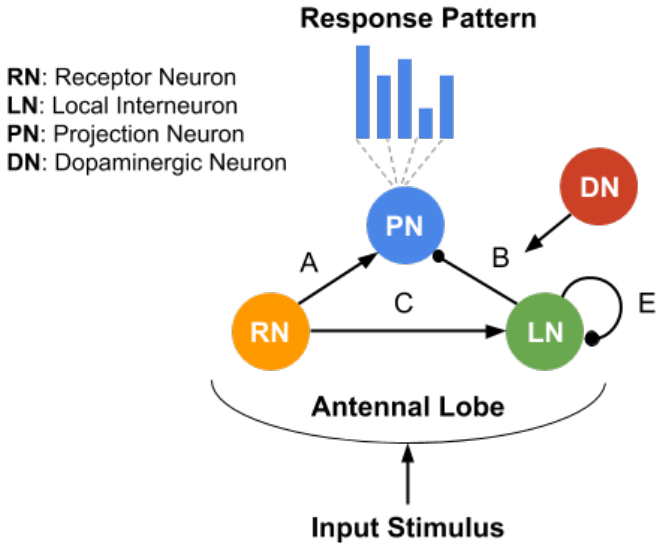

Figure 3. Network schematics of neurons in the antennal lobe. Both PNs and LNs receive input from RNs. RNs are connected to LNs and PNs mainly via inhibitory synapses. DA neurons in the model enhance synapses between the different neural bodies and modulate the responses of the individual PNs.

2.2.2 Neural Network Calibration to Pre-Dopamine State

We simplify the dynamics equation (Eq. 3-5) with two assumptions. First, the response of receptor neurons is locked onto the input J when there is significant input. Specifically, the activities of RNs converge to a fixed point ($x_0 = J$). Second, the inhibitory neurons respond faster than the excitatory neurons ($z_0 = (E + \gamma I)^{-1}CJ$), which leads to the deterministic equations of projection neuron over time that combines the connectome matrices:

$$\frac{dy}{dt} = -\beta y + (A - B(E + \gamma I)^{-1}C)J \quad (6)$$

Before we apply the learning rule to the lateral inhibition dynamics equation, we first calibrate the system of equations in P1 by using the data-driven top-down approach. Given that the firing rate patterns of the PNs during the first phase maintain stability, we assume that the connectome matrices A and B are time-invariant (no learning occurs). Thus, from the recorded firing rates of PNs, we calibrate the matrix B while fixing A, E, C, J for all odors.

We solve B using CVX, a package for solving convex programs in MATLAB (Grant and Boyd 2013). When specifying the convex problem, we constrained the weights of matrix B to be positive (**Figure 4** top right). Moreover, we calibrate the activity of RNs, x, for a more accurate depiction of the neuronal data and replaced J with x. Finally, we validate the calibrated parameters by computationally recovering the trajectory of PNs and their associated fixed points for each odor (**Figure 4**).

2.3 Computational Investigation of Dopamine Modulation

We hypothesize that for a given input x from the receptor neurons (fixed for all three phases), the calibrated system will successfully simulate the firing pattern of PNs using the following model equation:

$$\hat{y}(t) = (A - B(E + \gamma I)^{-1}C)x(t) \quad (7)$$

The element $B_{i,j}$ represents the connection weights from the local interneuron (LN) i to the projection neuron (PN) j. This gives direct insights into how dopamine modulates the synaptic strengths between LNs and PNs after the presentation of an odorant. Other connectivity matrices ($A, C,$ and E), as well as the calibrated input $x(t)$ are fixed throughout the iterations.

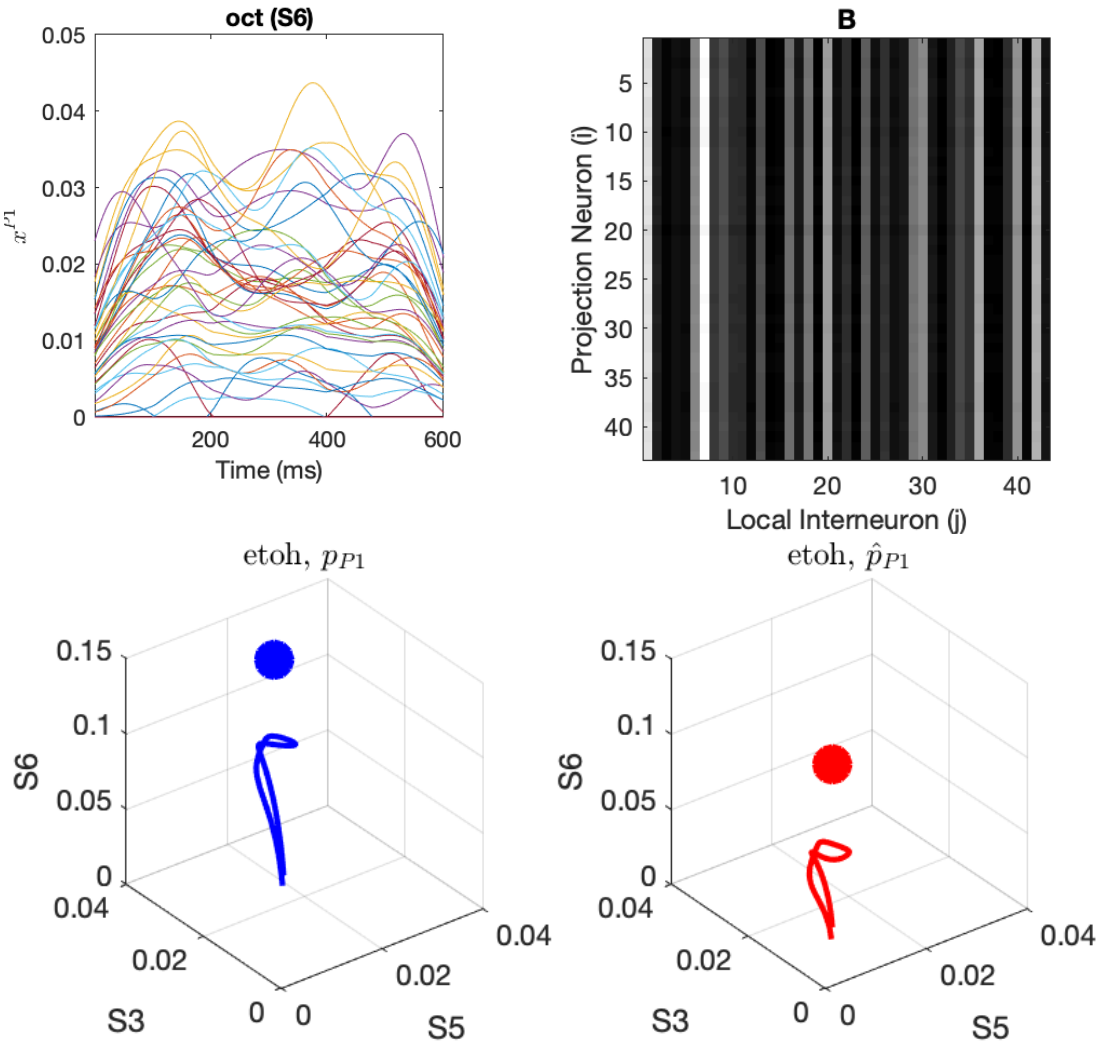

Figure 4. Top Left: example activities of each Receptor Neuron (RN) in response to Octenol over time in P1. Top Right: calibration of lateral inhibition matrix B using observed firing rates in P1. Bottom: the average of observed trajectories over trials (blue) and calibrated (red) trajectory and a fixed point in response to ethanol in P1.

Connectivity matrices and odor-dependent input x are calibrated from recorded data in P1. Learning is defined as an update to the connection matrix B after each trial n. The error e is used for connection updates by the learning method:

$$B_{i,j}(n) = B_{i,j}(n-1) + \eta e_{i,j} \quad (8)$$

During training, we define the loss as the mean absolute distance between the experimentally observed ('observed') firing rates and the simulated firing rates. Then, we sum the difference over K neurons:

$$J(n) = \sum_{\mathrm{K}} \frac{1}{\mathrm{T}} \sum_{\mathrm{t}} |\mathrm{y_j(t)} - \hat{\mathrm{y}}_\mathrm{j}(\mathrm{t})| \quad (9)$$

$y_j(t)$ and $\hat{y}_j(t)$ represent the observed and simulated firing rates of neuron j at time t, respectively.

We introduce three different learning rules to define the error. First, the error feedback rule simply computes the difference between the observed activities and the simulated activities of PNs. Next, we calculate the gradient of the loss with respect to B. To ensure convexity, we define the loss of the gradient as the L2-norm of the difference between the observed and computed PNs activities instead of the L-1 norm.

Another method introduced is Hebbian learning, a more biologically plausible approach. We update the weights of B by the accumulating the contributions of LNs to the change of PN activities.

With selected models, we also test if the batch type affects learning. During the in-vivo experiment, each neuron was presented with the same odor four times in the following order: oct (S1), la (S2), myr (S3), benz (S4), ammonia (S5), and etoh (S6). First, we train the model with the same conditions that were conducted in the experiment. Then, we repeat the training with trials selected at random. There are 24 possible trials (6 odors, four trials each) and control experiment is excluded from the training. For both conditions, the connectivity matrices are initialized with pre-calibrated ones at the start of the training.

2.3.1 Difference Error Feedback Rule

We first update the weights of the inhibition matrix B by the difference between the expected and observed firing rates. Explicitly, we define the error as:

$$\mathrm{e_j} = \frac{1}{\mathrm{T}} \sum_{\mathrm{t}} \left(\mathrm{y_j(t)} - \hat{\mathrm{y}}_\mathrm{j}(\mathrm{t}) \right) \quad (10)$$

Here, $\mathrm{e_j}$ is the error given to the column vector of B where each column vector $\mathrm{B_j}$ represents the weight connections to projection j. The error sum the difference between the computed firing rates ($\hat{\mathrm{y}}_\mathrm{j}$) and the desired firing rate ($\mathrm{y_j}$) over time T during trial n.

Figure 5 shows the recovery of the trajectory of PNs projected onto odor classification space following the update using three different learning rates η. The accuracy of learning is measured using the loss function (Eq. 9). The control plot indicates a loss with learning rate fixed to zero such that there is no learning occurring in the system. We observe that when the learning rate is too big ($\eta = 1.0$), the loss starts to diverge at trial 20 (**Figure 6**). When the learning rate is decreased, the loss does not diverge; however, the effect is minimal.

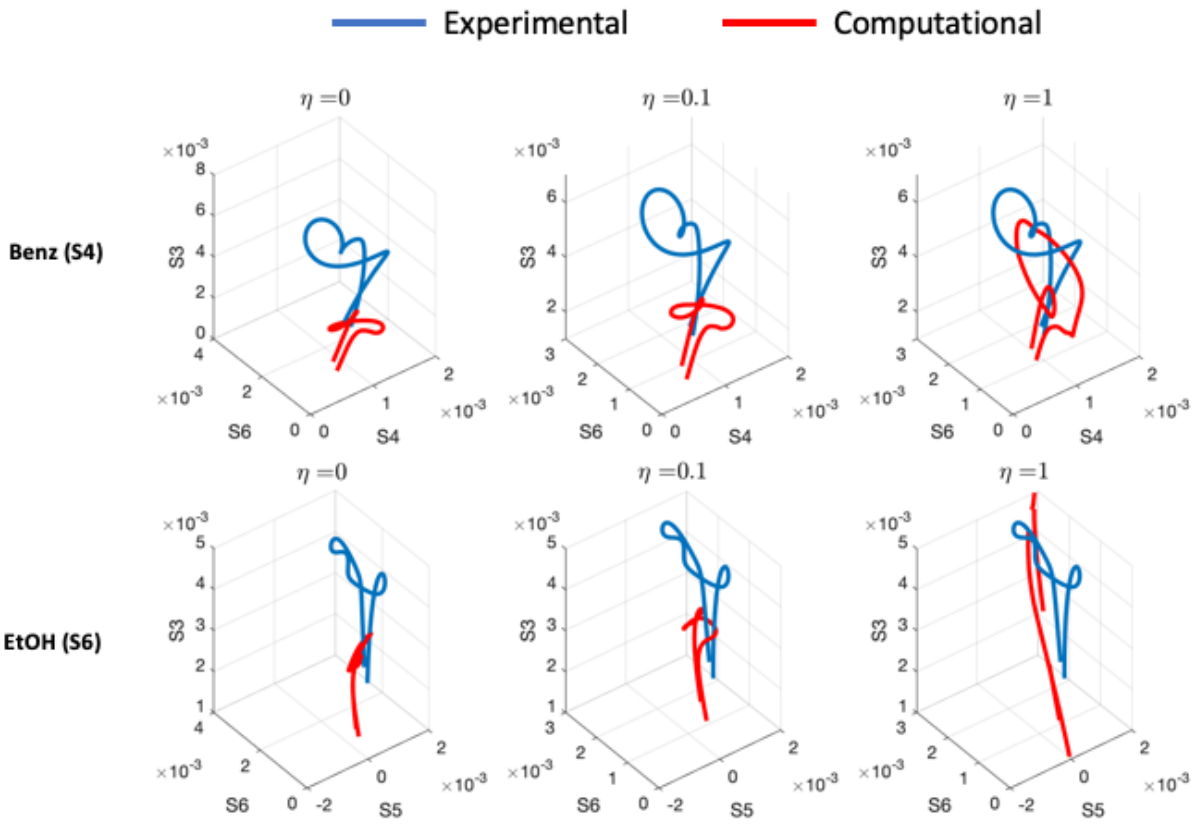

Figure 5. Recovery of the trajectory during training using three different learning rates (0, 0.1, 1.0) after the update on the last trial of Benz (S4), Ammonia (S5) and EtOH (S6).

2.3.2 Gradient Descent

Similarly, we apply the gradient descent rule using the cost function that sums the squared difference between the observed firing rates and the computed firing rates of the recorded neurons at each time point t. Given the new cost function:

$$\mathrm{J(t)} = \frac{1}{\mathrm{T}}\sum_{\mathrm{t}}(\hat{\mathrm{y}}(\mathrm{t}) - \mathrm{y(t)})^2 \tag{11}$$

we take the derivative with respect to B and compute the gradient:

$$\frac{\partial}{\partial \mathrm{B}}\mathrm{J(t)} = -\frac{1}{\mathrm{N}}\sum\big(\hat{\mathrm{y}}(\mathrm{t}) - \mathrm{y(t)}\big)B(E + \gamma I)^{-1}C)x(t) \tag{12}$$

$$\mathrm{e(t)} = \mathrm{e(t-1)} + \frac{\partial}{\partial \mathrm{B}}\mathrm{J(t)} \tag{13}$$

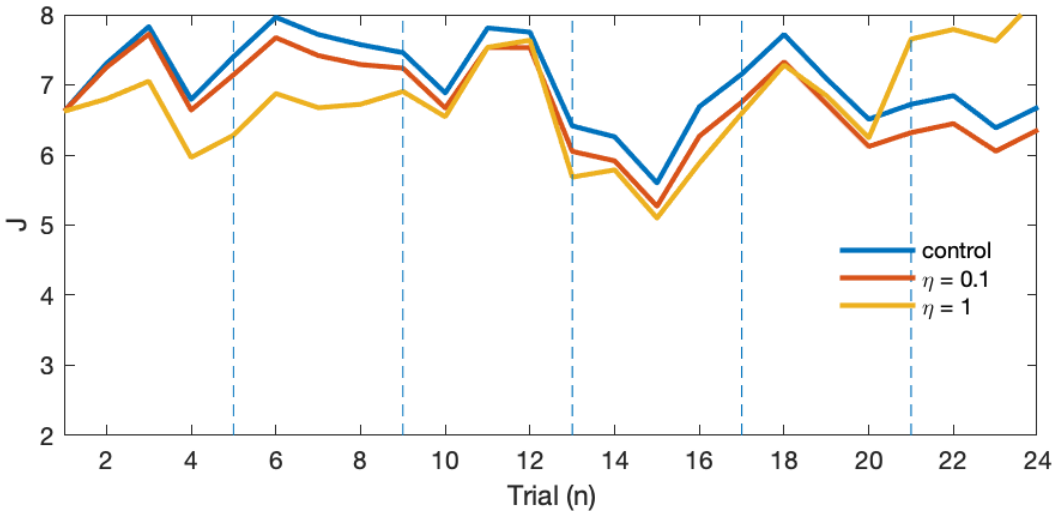

Figure 6. Training loss using three different learning rates (0, 0.1, 1.0) and difference error feedback rule. Blue dashed line indicates a switch to a new order.

The error, e, sum the gradient over time T. Similar to the difference error, we first initialize the network with the calibrated connectivity matrices, and after each trial, the error is added to B (Eq. 8). **Figure 7** shows the example dynamics of PNs after an update for the selected odors. Different learning rates are tested to find the optimal accuracy. When the learning rate is too small, the trajectories minimally change from P1. The learning rates are increased to 0.1 and 0.4. Although the learning rate of 0.4 generates more successful prediction with some odors like benzaldehyde, the result is less stable on others (i.e., ethanol).

To present a direct comparison to the difference error rule, we compute the accuracy of the model using the loss function in Eq. 9. **Figure 8** top shows the loss using the different learning rates compared to the control experiment (learning rate equals to zero, no updates of B). Learning rate of 0.4 successfully generates the expected trajectory for some odors but results in a higher loss in later trials (ammonia and ethanol).

Besides, using the learning rate that resulted in the least average loss ($\eta = 0.1$), we repeat the training using the trials selected at random without replacement. That is, the desired PNs activity, $y_i(t)$, is taken from responses to a random odor of random trial. **Figure 8** bottom compares the results of in-order and random batch types. Although random batch results in a smaller loss at the start of the training, we observe a few spikes in the loss throughout the trials and eventually converge to the loss in the control setting.

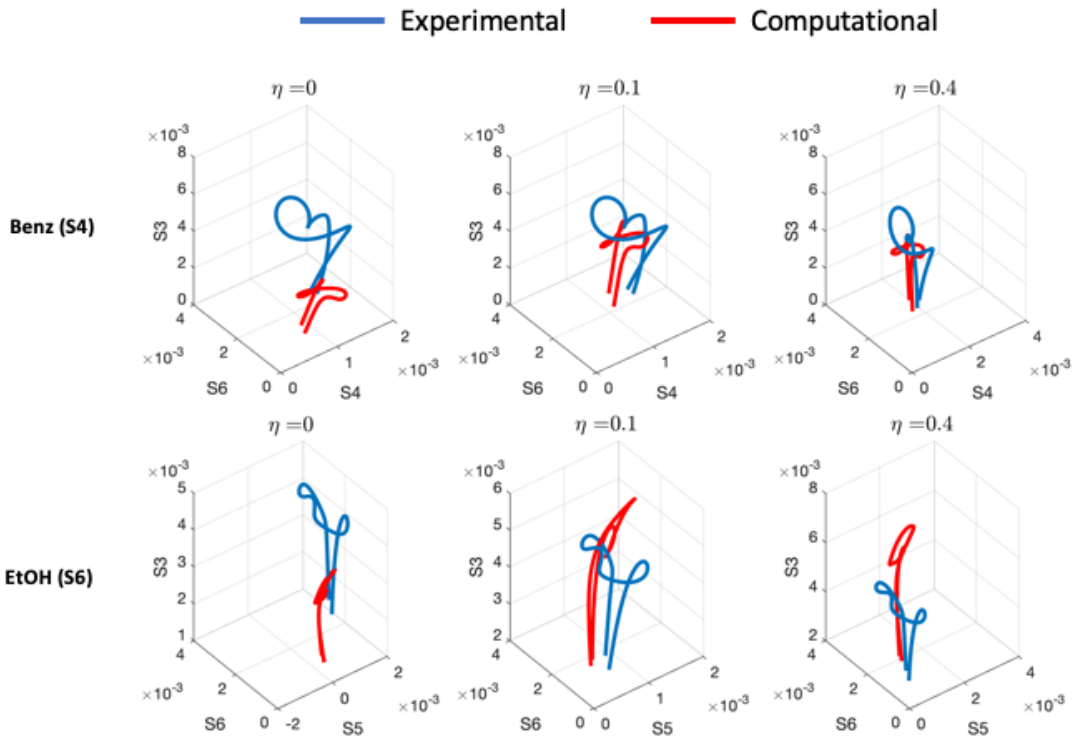

Figure 7. Recovery of the trajectory during gradient descent training using three different learning rates (0, 0.1, 0.4).

2.3.3 Hebbian Learning

Using Hebbian learning, we update the connection weights by the potential weights between two synapses over time. For our study, we multiply the activities of LNs (presynaptic neuron) at the

previous time step and the connected PNs (postsynaptic neuron) at the current step. To assign negative and positive signs to the presynaptic activity, we add the activation function to the output of LNs, modifying the model equation (Eq. 7):

$$\hat{y}(t) = Ax(t) - Br(t) + dy(t) \quad (14)$$

$$r_i(t) = \tanh\left((E + \gamma I)^{-1}C\right)x(t)) \quad (15)$$

$r_i(t)$ represents the output of local interneuron i and thus the current input to projection neuron j. $dy(t)$ represents random perturbation to projection neurons centered at 0.01.

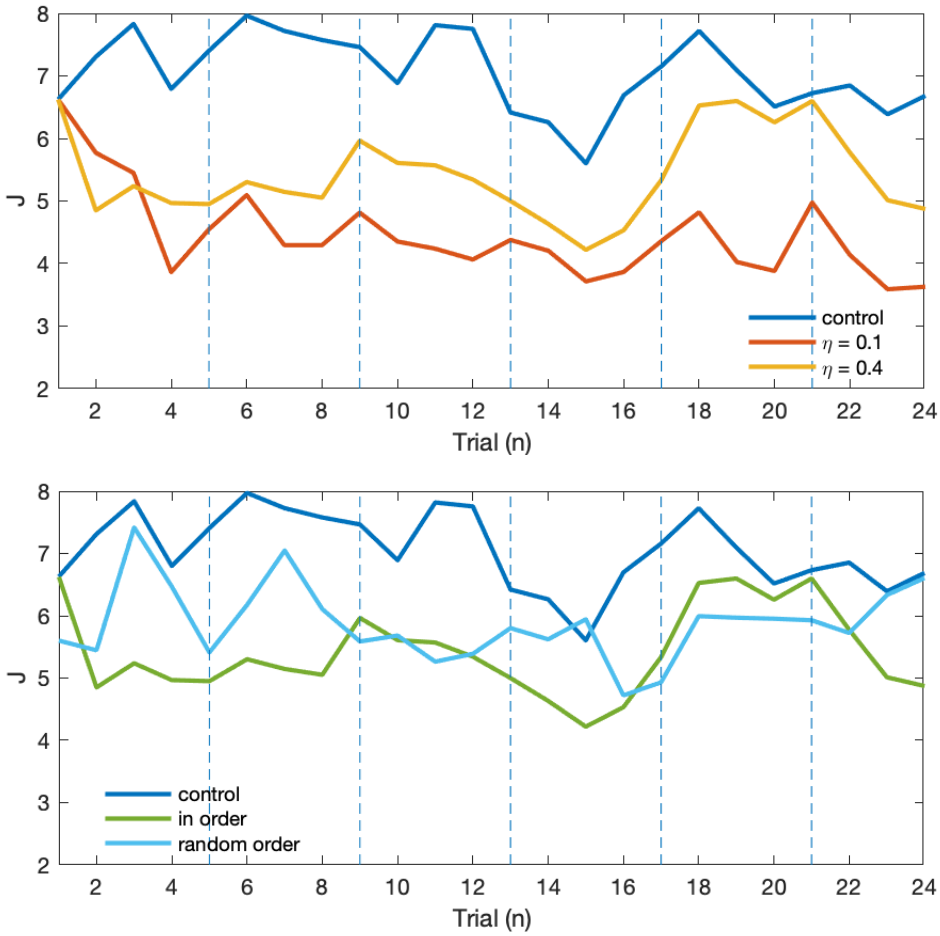

Figure 8. (Top) Loss over trials during gradient descent training using three different learning rates (0, 0.1, 0.4). (Bottom) Gradient descent method with in-order and pseudo-random training sets.

Then we define the error signal as follows:

$$e_{i,j}(t) = e_{i,j}(t-1) + r_i(t-1) * (y_j(t) - \bar{y}_i) \quad (16)$$

$$\bar{y} = \alpha\bar{y} + (1-\alpha)y(t) \quad (17)$$

$\bar{y}$ represents a short-term running average of y_j observed in P2. Thus, $y_j(t) - \bar{y_i}$ measures the deviation of the projection neuron i from its previous activities. $e_{i,j}$ sums the potential contribution of the local neuron j to the trace of the projection neuron i over time. After each trial, the inhibition matrix B is updated according to Eq. 8.

Similar to two previous methods, Hebbian learning is tested with three different learning rates, where the learning rate of zero served as the control (**Figure 9**). We observe that with Hebbian learning, the shape of the trajectory is more conserved from P1.

Figure 10 shows the effects of learning rate and batch type on the PN firing patterns during training. The learning rate of 0.25 displays similar patterns shown with gradient descent with a learning rate of 0.4. With Hebbian learning, the learning rate of 0.4 results in highly unstable trajectories, which indicates that a more careful selection of learning rate is required for Hebbian learning. On the other hand, similar to gradient descent, random batch type leads to a sudden increase in loss on some trials, but no divergence is observed.

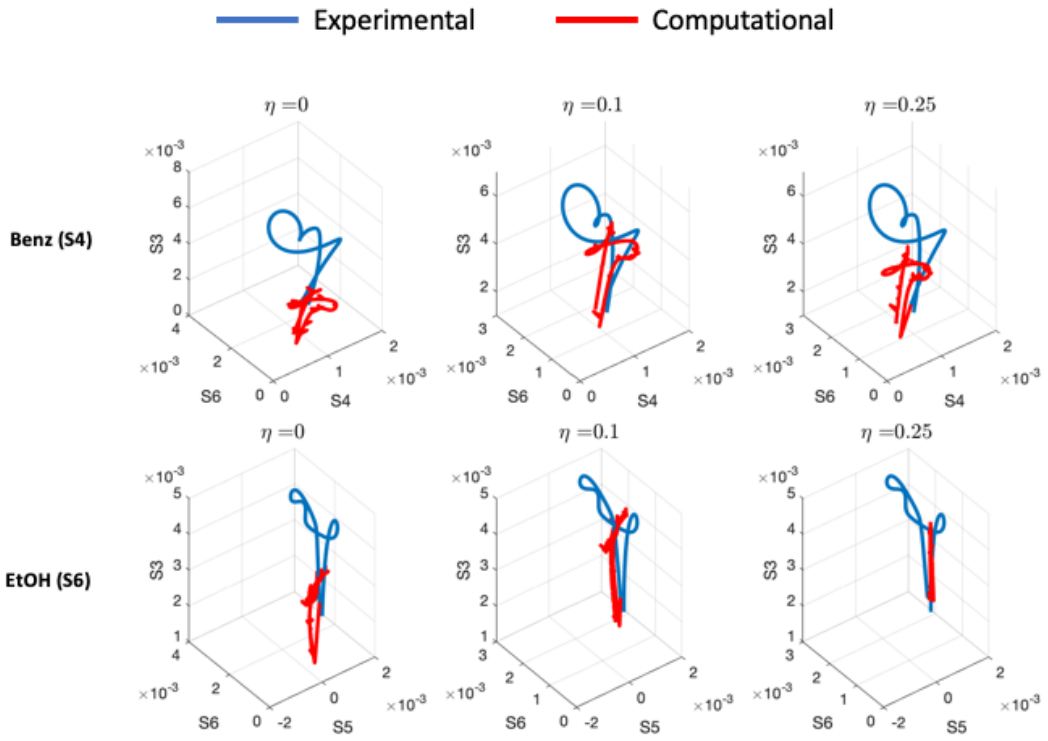

Figure 9. Training of Projection Neurons using Hebbian Learning and three different learning rates (0, 0,1, 0.25). Each trajectory uses the updated B to simulate the dynamics of PNs.

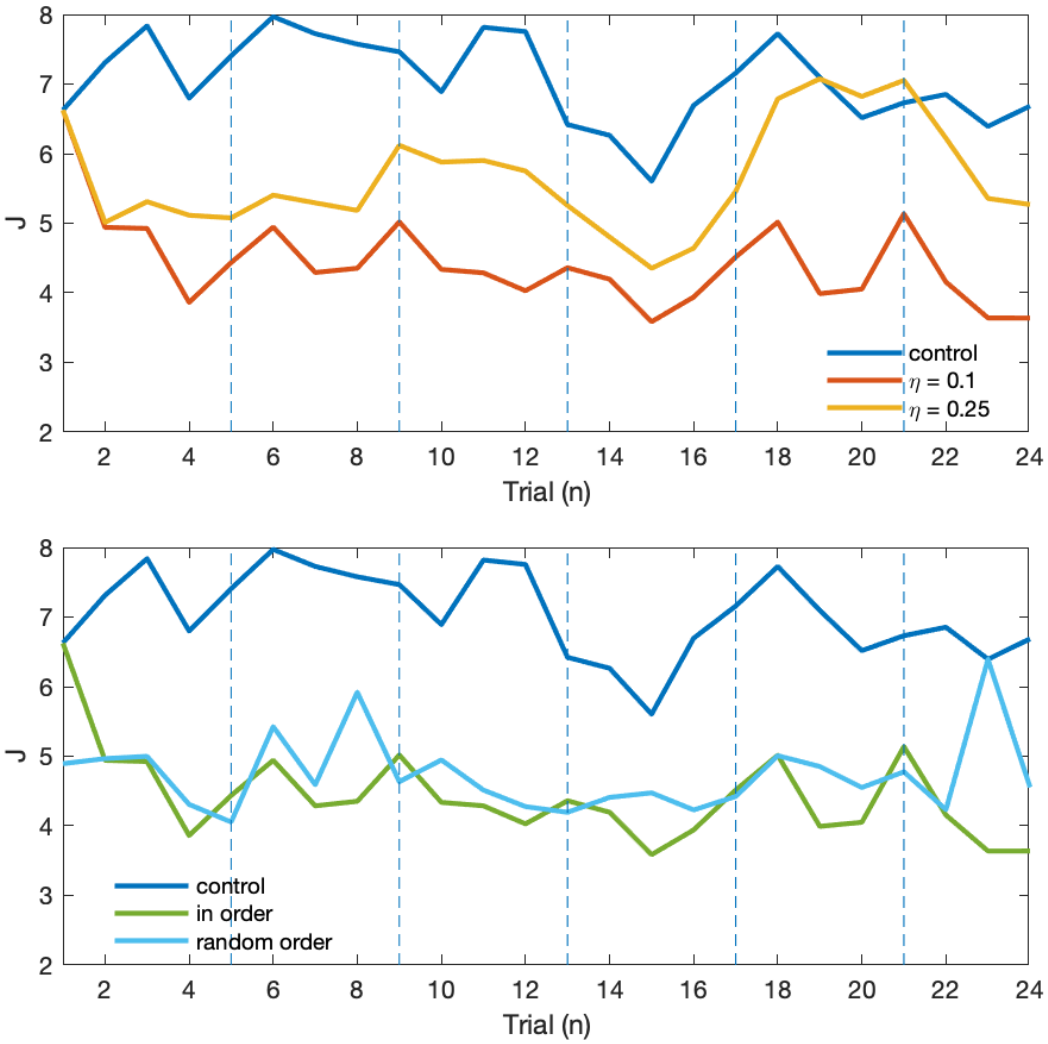

Figure 10. (Top) The loss during training using Hebbian Learning rule. The control curve represents the learning rate of zero. (Bottom) Hebbian Learning with in-order and pseudo-random training sets.

2.3.4 Reward-Based Hebbian Learning

Gradient descent and Hebbian learning display effective recovery of firing patterns in P2 through the updates of the inhibition matrix B. However, the above methods do not explain the basis of learning, and how dopamine mediates it. To do that, we define a dopamine-related reward signal that depends on the biological reward and predicted reward.

Since there was no artificial reward given throughout the experiment, we assume that the external application of dopamine only altered the prediction itself, not the reward in P2. Thus, we

expect the true reward predicted by the firing patterns of P1 to be intact throughout the experiment. Based on this assumption, we define the reinforcing signal and replace the firing patterns of P2:

$$R(t) = \bar{y}^{P1} - \hat{y}(t-1) \quad (18)$$

$$e_{i,j}(t) = e_{i,j}(t-1) + R(t)|r_i(t-1)| \quad (19)$$

$\bar{y}^{P1}$ is the mean firing rates over time T averaged over four trials. The difference between the simulated firing patterns and the true prediction signal serve as the reinforcing signal. Then, we propagate the magnitude of reinforcing signal to the output of the local interneuron, which is defined by $r(t)$. Here, we only take the sign from the reinforcing signal by taking the absolute value of $r(t)$.

The plot in **Figure 11** shows that the reward-based Hebbian Learning decreases the error between the observed and the computed firing patterns over trials. Compared to other methods, the minimum loss is achieved at trial 15 with $J = 4.418$. Although this value is comparable to results produced by other methods, we find that reward-based Hebbian learning requires more trials to reach the minimum loss.

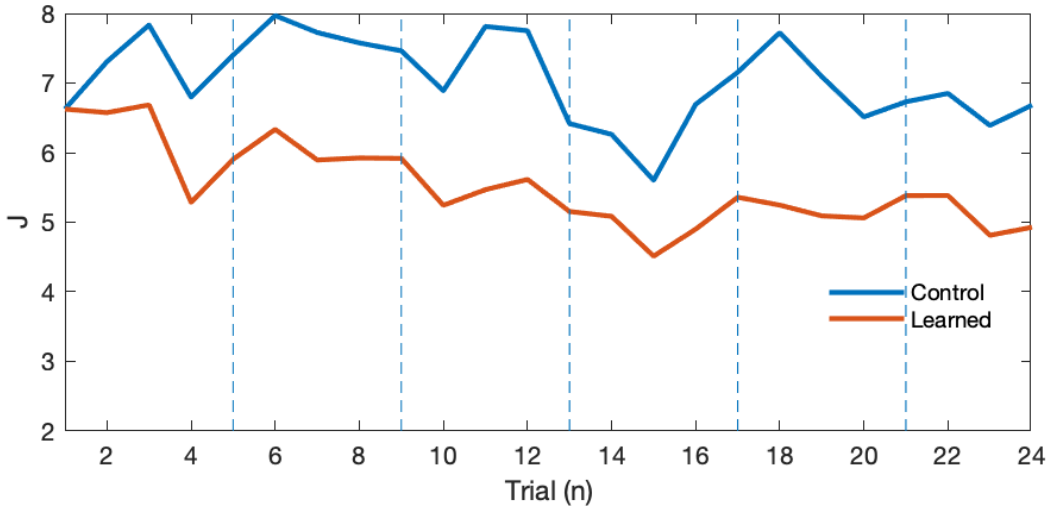

Figure 11. The loss during training using reward-based Hebbian Learning rule. The control loss represents the learning rate of zero.

2.4 Evaluation of the Tested Models

Table 3 summarizes the training error for each odor and the average for the tested learning rules. Control error serves as the benchmark for comparison—the lateral inhibition matrix B is not updated and sets the expected train error without the learning. For average accuracy, we take the

mean of errors reported from trial 17 to 24. We find that the Gradient Descent method is preferred over difference error and that Hebbian Learning produces comparable results. Hebbian reward, on the other hand, performs better than the difference error, but the training loss is higher than that of the gradient descent and Hebbian learning.

Table 3. Train error using the fixed points in P2.

	Control	Difference Error	Gradient Descent	Hebbian Learning	Hebbian Reward
Octenol (S1)	6.795	6.749	3.863	3.861	5.285
Lactic Acid (S2)	7.575	7.398	4.294	4.353	5.924
Myrcene (S3)	7.754	7.610	4.064	4.029	5.615
Benzaldehyde (S4)	6.692	6.541	3.865	3.940	4.900
Ammonia (S5)	6.516	6.266	3.881	4.054	5.061
Ethanol (S6)	6.687	6.465	3.628	3.639	4.927
Average	**6.893**	**6.672**	**4.177**	**4.268**	**5.158**

To test whether the learned connectivity is fixed after the learning phase of P2, we attempt to predict the firing patterns in P3 by fixing the learned B from P2. As a control, we fix the matrix B as in P1. **Figure 12** compares the difference between the observed and predicted firing patterns in P3 using the learned matrix B in P2. We also test if a continuation of learning can further decrease the loss.

Notably, the updated matrix B from P2 results in a much smaller loss in the first half of the test trials (1-12) in comparison to the control. However, as the trials proceeded (13-24), the loss computed from the updated B (B_{P2}) converge to the control one (B_{P1}).

We, therefore, test if learning continues throughout P3 as well, i.e., we apply the same update rule during P3 (yellow curve in **Figure 12**). We observe that the update rule further decreases the error over trials. **Table 4** summarizes the errors per odor applying the same learning rule used for P2. The decrease in test loss reveals that learning persists during P3. In addition, updating the connection weights of B using Hebbian learning appears to best predict the firing patterns of P3.

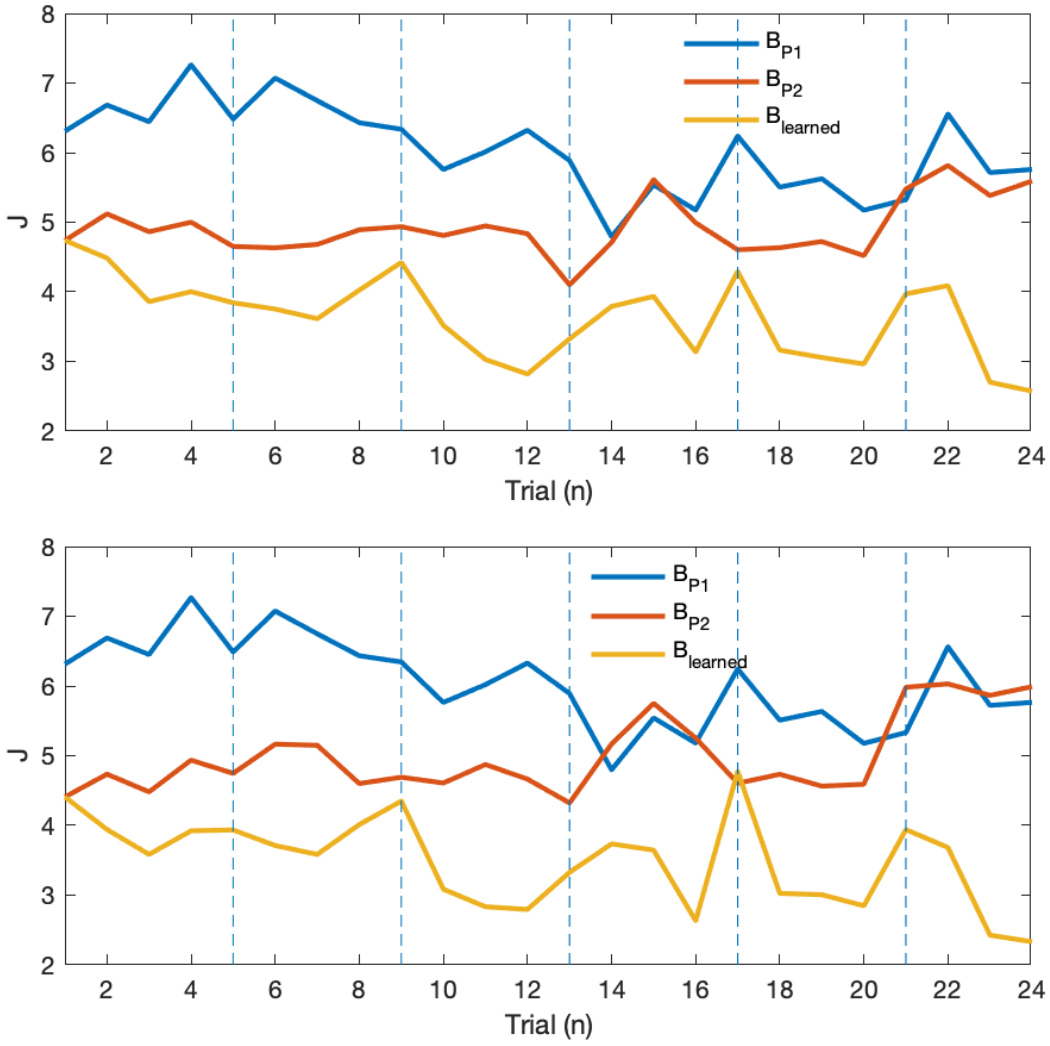

Figure 12. (Top) the loss during testing using Gradient Descent. (Bottom) Test loss using Hebbian learning. First, the firing patterns are predicted with calibrated B from P1 (B_{P1}, blue). The firing patterns are then predicted with fixed B taken from P2 after training trials (B_{P2}, red). The process is repeated with B initialized with B_{P2} and ongoing update rule ($B_{learned}$, yellow).

Table 4. Loss during the test phase (P3) using fixed B from different states.

Stimulus Name & Label	B_{P1}	B_{GD}	B_{HL}
Octenol (S1)	7.261	4.003	3.920
Lactic Acid (S2)	6.429	4.024	4.022
Myrcene (S3)	6.321	2.818	2.784
Benzaldehyde (S4)	5.176	3.135	2.635
Ammonia (S5)	5.173	2.963	2.826
Ethanol (S6)	5.758	2.572	2.346
Average	**5.737**	**3.351**	**3.238**

2.5 Comparative Weights of the Inhibition Matrix

Using the elements of B in three different phases, we compare how the inhibition matrix affects the projection neurons under dopamine superfusion and washouts. More specifically, by summing the column vectors of B, we measure the contribution of LNs to each PN. **Figure 13** shows the comparative weights between the two phases by subtracting the summed column vectors of B. $B_{P3}$is generated with the continued update learning rules under test trials. Therefore, each row in the bar plots shows the overall change in the activity in each projection neuron j in P2 and P3.

For both methods used, the inhibition connections are generally weaker in P2 and stronger in P3. Although the resulting B matrices from two methods are different element-wise, both plots show that when summed together, both methods display a similar pattern. For example, neurons that are strongly inhibited in P2 (neuron 8-11) recover their weights in P3 under gradient descent and Hebbian learning.

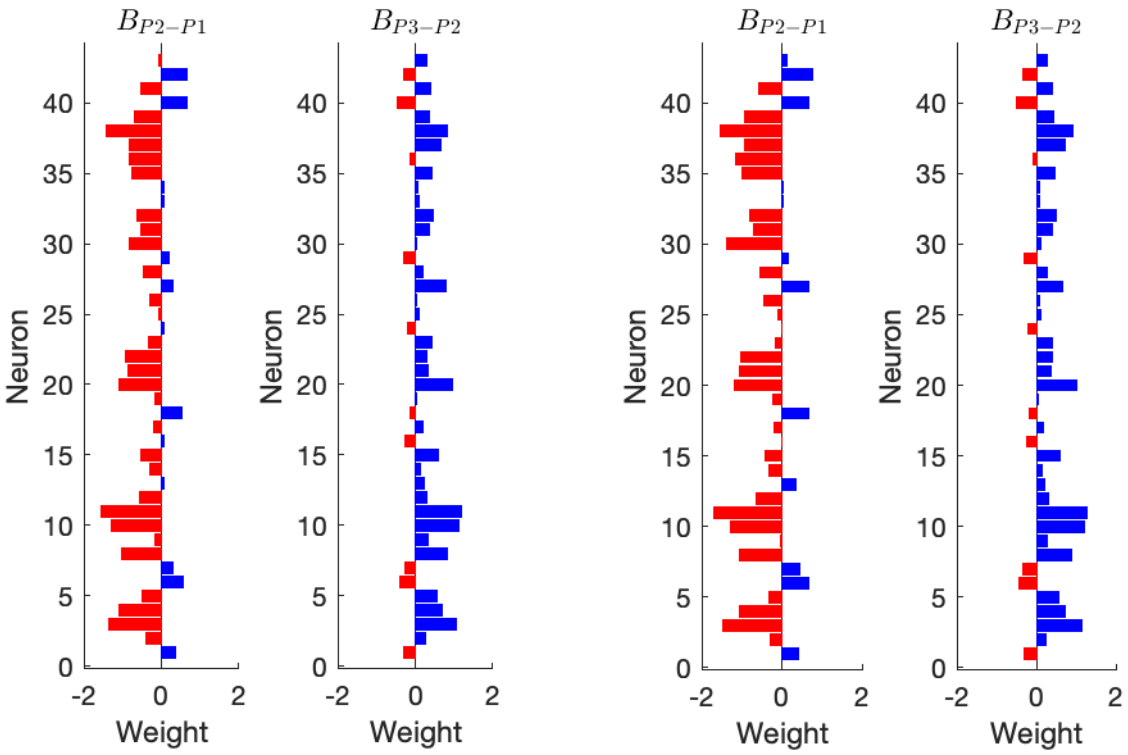

Figure 13. Comparative weights of B updated with Gradient Descent (left) and Hebbian learning (right). The red bar indicates the decrease in weight and the blue bar indicates the increase in weight compared to the previous phase.

3. Discussion

Using a top-down approach, we develop a data-driven network model for the neural codes that appear in the extracellular recordings, from the AL projection neurons. The proposed model shows that the role of dopamine neurons can be represented by extending the lateral-inhibition technique and updating the lateral inhibition connection between local interneurons and projection neurons.

The essence behind this study is to investigate the influence of dopamine on the AL neural activity in the mosquito brain. The dynamics of PNs projected onto the lower dimensional odor space reveals how time-varying firing patterns of each neuron reach its maximal value in response to different odorants. Comparison of the locations of the fixed points in each of the phases indicates that the superfusion of dopamine causes fixed points to dislocate signifying that dopamine has long-term effects on the AL neural encoding (**Figure 2**). Furthermore, dislocation of the fixed point appears to be specific for each odor (**Table 2**).

Using the model that employs lateral inhibition, we use the trajectories observed in P1 to calibrate the parameters of the lateral inhibition model. To reduce the number of unknowns, we assume one-to-one excitatory synapses between RNs and PNs, and between LNs and PNs (connection matrices A and C are fixed as identity). Under these conditions, the lateral inhibition B is solved with 16 iterations using the standard semidefinite programming.

As shown in **Figure 4** bottom, the trajectories do not reach its fixed point in a given trial since the fixed point location is calculated with the maximal activity from all four trials. This is also reflected in the computationally simulated trajectory that employed the pre-calibrated parameters (**Figure 4** bottom right). The successful recovery of both the time-varying firing patterns and the fixed points in P1 suggests the assumptions discussed above are valid in P1.

With the model trajectories and calibrated parameters, we experiment how the connectomes interacted to mediate the response of the PNs. Specifically, we use three supervised learning methods: simple difference error method, gradient descent, and Hebbian learning. The best training accuracy is obtained through gradient descent and Hebbian Learning. These learning rules demonstrate that updating matrix B can successfully generate short-term trajectories under dopamine modulation (**Figure 7**, **Figure 9**).

Also, we replace the constant supervisory signal with the firing rates pattern in P1 using the reward-based Hebbian learning. Although the training loss in **Figure 11** is suboptimal compared to the learning rules tested above, it suggests that using firing rate patterns in P1 as a reinforcing signal offers a reasonable prediction.

Figure 12 shows test errors that converge toward an even lower value than in P2. It suggests that the network undergoes learning even after dopamine washout. Thus, it further proves that the activities of DA neurons are mediated by the extracellular concentration of dopamine as well as the error between the actual reward and predicted reward. Since the experimental setup did not include an external reward that matched the external modulation of dopamine, the system needs to recalibrate in P3.

Finally, the comparative weights in **Figure 13** demonstrate the lateral inhibition matrix after learning in P3 produces the connection weights opposite to those seen in P2. Even without the external application of dopamine, the synaptic strengths constantly change to signal the error between expected and actual reward. This reveals that the observed reward does not fully match prediction in P3.

In contrast to the overall recovery pattern of matrix B, fixed point of each odor does not recover back to its original position in P3 (**Figure 2**). Thus, fixed point relocation emphasizes that dopamine had long-term effects on PN firing patterns for each odorant, and DA neurons in P3 did not reverse the learning occurred in P2. Taken together with the recovery pattern of matrix B, this observation illustrates how DA neurons accumulate odor-dependent modulations over trials, which eventually leads to the full recovery of the original connectivity in P3.

Furthermore, we test if the order of learning procedure has a significant impact on reorganizing the lateral inhibition matrix B. **Figure 13** shows that the batch type does affect the loss. Training set with random trials results in a higher loss overall, yet its effects are not sufficient to offset the learning itself.

The only limitation of the tested model is that it did not explore the modulation on excitatory synapses. Devising a cost function for such a problem is not be trivial because the excitatory responses are much slower than the inhibitory responses. Assigning weight changes to one of these neurons may potentially change the future activity of other neurons.

4. Conclusion

Our primary contribution is the addition of DA neural connections to the lateral inhibition model, with the update rule that modulates the synaptic weights between LNs and PNs. Using supervised learning methods (gradient descent and Hebbian learning), we simulate the firing rate patterns of PNs under dopamine modulation. On studying the comparative weights of the modulation matrix D, we can conclude that the DA neurons assist in recovering the overall contributions of LNs to PNs after all tested trials. Also, the discovery of trial-dependent modulation suggests that the dynamics of odor recognition rely on the predefined inhibitory connectomes and the strengths of the DA neurons varying over different odorants.

References:

Arevian, Armen C, Vikrant Kapoor, and Nathaniel N Urban. 2008. "Activity-Dependent Gating of Lateral Inhibition in the Mouse Olfactory Bulb." *Nature Neuroscience* 11 (1): 80–87. https://doi.org/10.1038/nn2030.

Berke, Joshua D. 2018. "What Does Dopamine Mean?" *Nature Neuroscience* 21 (6): 787–93. https://doi.org/10.1038/s41593-018-0152-y.

Blaszka, David, Elischa Sanders, Jeffrey A. Riffell, and Eli Shlizerman. 2017. “Classification of Fixed Point Network Dynamics from Multiple Node Timeseries Data.” *Frontiers in Neuroinformatics* 11: 58. https://doi.org/10.3389/fninf.2017.00058.

Chilaka, Nora, Elisabeth Perkins, and Frédéric Tripet. 2012. “Visual and Olfactory Associative Learning in the Malaria Vector Anopheles Gambiae Sensu Stricto.” *Malaria Journal* 11: 1–11. https://doi.org/10.1186/1475-2875-11-27.

Grant, Michael, and Stephen Boyd. 2013. “CVX: Matlab Software for Disciplined Convex Programming, Version 2.0 Beta.” September 2013. http://cvxr.com/cvx.

Hirsch, Ja, and Cd Gilbert. 1991. “Synaptic Physiology of Horizontal Connections in the Cat’s Visual Cortex.” *The Journal of Neuroscience* 11 (6): 1800–1809. https://doi.org/10.1523/JNEUROSCI.11-06-01800.1991.

Hoerzer, Gregor M., Robert Legenstein, and Wolfgang Maass. 2014. “Emergence of Complex Computational Structures From Chaotic Neural Networks Through Reward-Modulated Hebbian Learning.” *Cerebral Cortex* 24 (3): 677–90. https://doi.org/10.1093/cercor/bhs348.

Joel, Daphna, Yael Niv, and Eytan Ruppin. 2002. “Actor–Critic Models of the Basal Ganglia: New Anatomical and Computational Perspectives.” *Neural Networks* 15 (4–6): 535–47. https://doi.org/10.1016/S0893-6080(02)00047-3.

Montague, Pr, P Dayan, and Tj Sejnowski. 1996. “A Framework for Mesencephalic Dopamine Systems Based on Predictive Hebbian Learning.” *The Journal of Neuroscience* 16 (5): 1936–47. https://doi.org/10.1523/JNEUROSCI.16-05-01936.1996.

Riffell, J. A., E. Shlizerman, E. Sanders, L. Abrell, B. Medina, A. J. Hinterwirth, and J. N. Kutz. 2014. “Flower Discrimination by Pollinators in a Dynamic Chemical Environment.” *Science* 344 (6191): 1515–18. https://doi.org/10.1126/science.1251041.

Schultz, Wolfram. 1997. “Dopamine Neurons and Their Role in Reward Mechanisms.” *Current Opinion in Neurobiology* 7 (2): 191–97. https://doi.org/10.1016/S0959-4388(97)80007-4.

Shlizerman, Eli, Jeffrey A. Riffell, and J. Nathan Kutz. 2014. “Data-Driven Modeling of the Olfactory Neural Codes and Their Dynamics in the Insect Antennal Lobe.” *Frontiers in Computational Neuroscience* 8 (August). https://doi.org/10.3389/fncom.2014.00070.

Soltoggio, Andrea, and Jochen Steil. 2013. “Solving the Distal Reward Problem with Rare Correlations.” *Neural Computation* 25.4: 940–78.

Suri, R. E., and W. Schultz. 1999. “A Neural Network Model with Dopamine-like Reinforcement Signal That Learns a Spatial Delayed Response Task.” *Neuroscience* 91 (3): 871–90. https://doi.org/10.1016/S0306-4522(98)00697-6.

Vinauger, Clément, Chloé Lahondère, Gabriella H. Wolff, Lauren T. Locke, Jessica E. Liaw, Jay Z. Parrish, Omar S. Akbari, Michael H. Dickinson, and Jeffrey A. Riffell. 2018. “Modulation of Host Learning in Aedes Aegypti Mosquitoes.” *Current Biology* 28 (3): 333-344.e8. https://doi.org/10.1016/j.cub.2017.12.015.