Supplementary information for Developmental and evolutionary constraints on olfactory circuit selection

Naoki Hiratani and Peter E. Latham

Supplementary figures

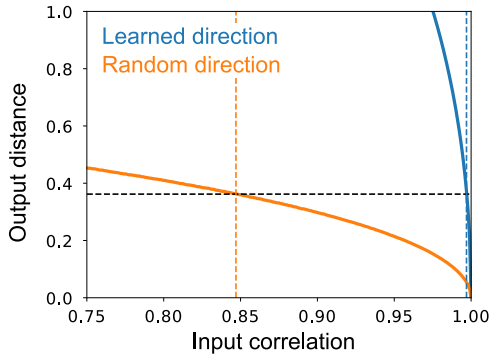

Figure S1: Valence discrimination after learning. The uncertainty in valence is $\sqrt{\epsilon_{gen}}$, so we assume that odors can be distinguished if their valances differ by this amount. The predicted valence, $\hat{y}$, is given in terms of odor, $\boldsymbol{x}$, by $\hat{y} = \boldsymbol{w}_s \cdot g_s(\boldsymbol{J}_s\boldsymbol{x})$ (Eq. (3)). For the distance between two odors we use the correlation coefficient. We consider two kinds of odor pairs: random and optimal. For random odors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we let $\boldsymbol{x}_2 = \rho\boldsymbol{x}_1 + \sqrt{1-\rho^2}\boldsymbol{\xi}$, where ρ, which lies between 0 and 1, is the correlation coefficient, and $\boldsymbol{\xi}$ and $\boldsymbol{x}_1$ are random Gaussian vectors whose components are zero mean, unit variance, and independent. For optimal odors, we let $\boldsymbol{x}_2 = \rho\boldsymbol{x}_1 + \sqrt{1-\rho^2}\boldsymbol{\xi}_{opt}$ where again $\boldsymbol{x}_1$ is a random Gaussian vector, but now $\boldsymbol{\xi}_{opt} = \boldsymbol{J}_s^T g'(\boldsymbol{J}_s\boldsymbol{x}_1) \odot \boldsymbol{w}_s$ is the direction with the maximum gradient under the learned weight $\boldsymbol{w}_s$ (here $\odot$ refers to element-wise multiplication: $(\mathrm{a} \odot \mathrm{b})_i = a_i b_i$). We plot the mean distance between the outputs, $\langle |\hat{y}_1 - \hat{y}_2| \rangle_{x_1,x_2}$, versus the correlation coefficient, ρ. Orange line: random input. Blue line: optimal input. The black horizontal line indicates the square-root of the generalization error, $\sqrt{\epsilon_{gen}}$. We used $L_x = 50$, $N = 30000$, and $\sigma_t^2 = 0.1$ as in Fig. 3, and set L_h to the numerically estimated optimal value, $L_h = 2924$.

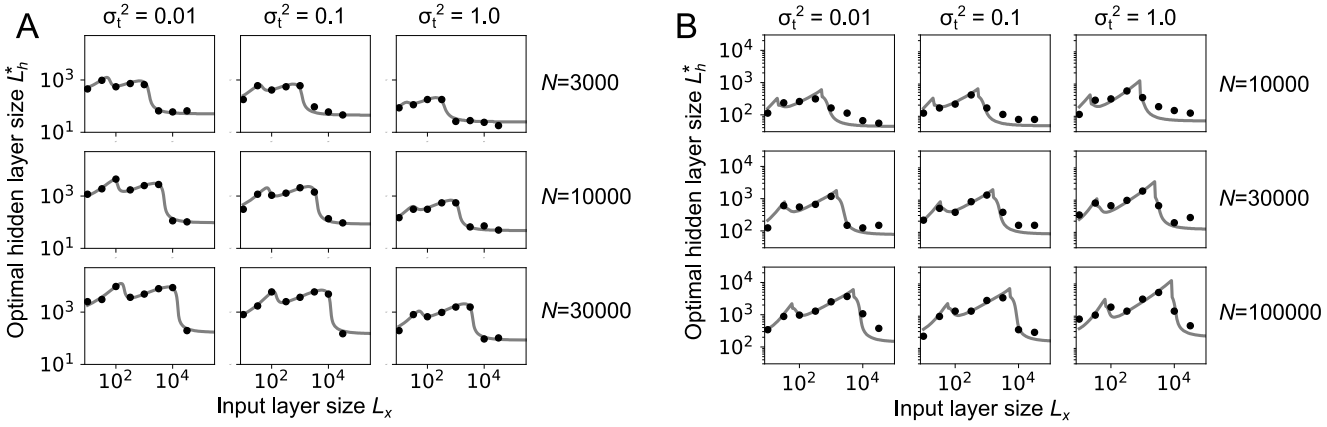

Figure S2: Optimal hidden layer size under various teacher noise levels, σ_t^2, and the number of samples, N. **A**) Maximum likelihood estimation (MLE). **B**) Stochastic gradient descent (SGD). In both panels, points are simulations and lines are theory, derived from Eq. (63) and Eq. (114), respectively. We used ReLU for the activation function of both teacher and student models, and in panel B we set the initial weight, σ_R^2, to 9.0. Under both MLE and SGD, the optimal hidden layer size L_h^* increases as the number of samples N gets larger. Under MLE, L_h^* also increases as the teacher noise σ_t^2 decreases, but L_h^* is mostly invariant with respect to σ_t^2 under SGD. When $N = 3000$ (top row in panel A), the second phase under MLE is relatively flat; that's because the lines saturate before the scaling fully kicks in.

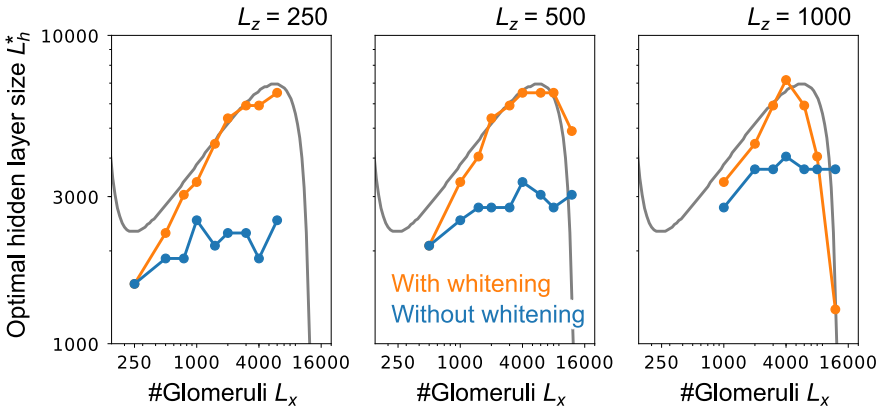

Figure S3: Scaling under input with a low-dimensional structure. In our model, we assume that different glomeruli are independent. This is a reasonable assumption when each glomerulus receives input from one type of olfactory receptor neuron (OSN). However, in mammals each OSN type projects to several glomeruli [1] while each glomerulus expectedly receives inputs from one OSN types [2]; in this case the glomeruli cannot be independent. To determine the effect of this low dimensional structure, we computed the optimal hidden layer size in this regime. We fixed the number of OSN types, denoted L_z, to 250, 500, or 1000, and estimated the optimal hidden layer size under different number of glomeruli, L_x (x-axis; see SI §7.6 for the details of the model). The results are shown for these three values of L_z, from left to right. Blue and orange lines are simulation results with and without whitening by lateral inhibition among glomeruli, respectively. The gray lines are the theoretical estimation in the absence of any low-dimensional structure ($\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I})$). Consequently, so long as the input to the glomeruli is whitened due to lateral inhibition, a salient feature of the olfactory bulb [3], the scaling we find for the hidden layer versus number of glomeruli should apply.

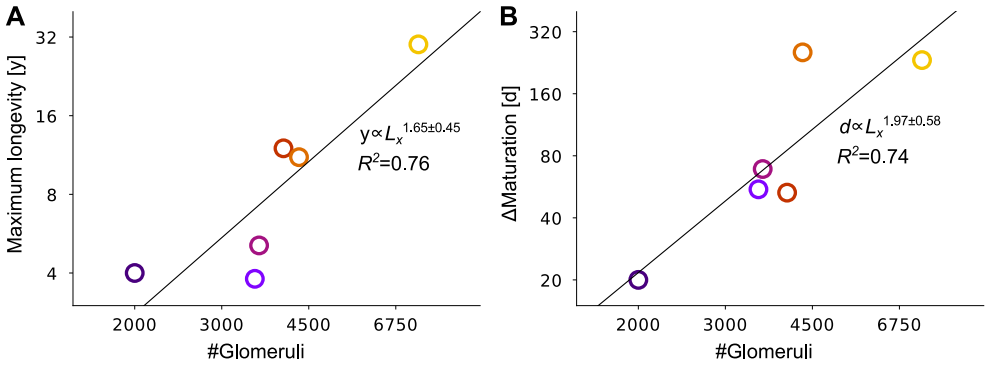

Figure S4: Scaling of duration of learning for vertebrates with the number of glomeruli. **A)** Maximum longevity versus number of glomeruli. **B)** Average duration from weaning to sexual maturation versus number of glomeruli. Color code is the same as in Fig. 1A.

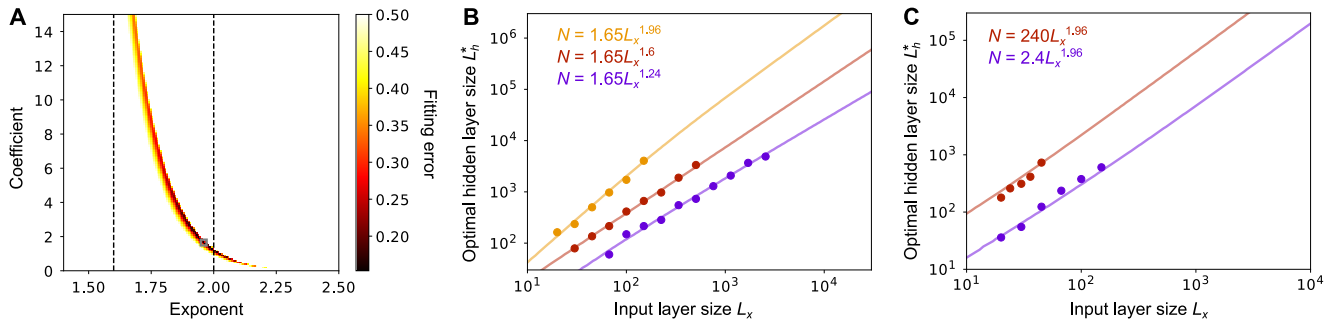

Figure S5: Sensitivity analysis and scaling between the optimal hidden layer size and the input layer size under maximum likelihood estimation. **A)** Fitting error under various choices of $N = CL_x^{\gamma}$, as a function of the exponent γ and the coefficient C, calculated from the analytical estimation of the optimal hidden layer size. Colorbar represents the normalized MSE $\sqrt{\frac{1}{6}\sum_i \left(L_{h,i}^{(model)}/L_{h,i}^{(data)} - 1\right)^2}$ over six data points $L_{h,i}^{(data)}$ in Fig. 1A. The gray square represents the parameter with the minimum error $(C, \gamma) = (1.65, 1.96)$ which we used in Fig. 4C. **B)** Scaling under various exponents, γ (which relates N to L_x via $N \propto L_x^{\gamma}$). The slopes of L_h^*-L_x scaling are estimated to be 1.13 (purple), 1.29 (red), and 1.41 (yellow), from linear regression on the theory curves. For the yellow line, we estimated the slope of the second phase. Both in panel A and B, the activation function of both student and teacher networks are ReLU. **C)** Scaling when the student activation function is the logistic function while the teacher function is ReLU. In both panels, points are simulations and lines are analytical results. The slopes are estimated to be 1.43 (purple), and 1.47 (red).

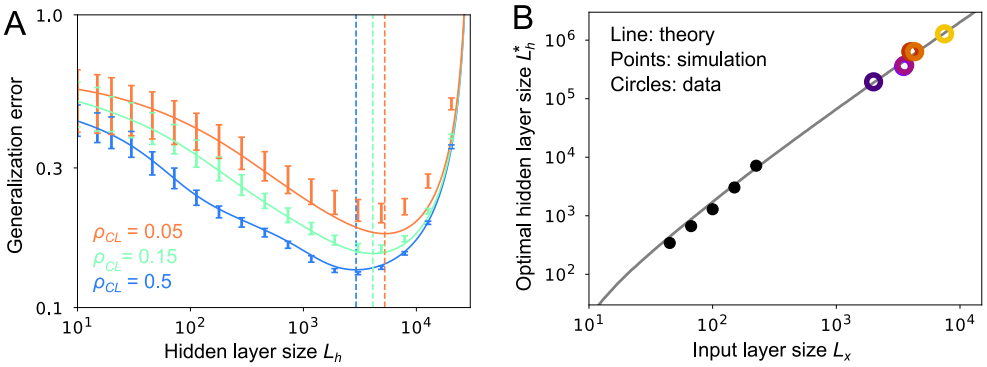

Figure S6: Scaling in a sparse coding model under MLE. **A)** The generalization error for three different coding levels, ρ_{CL} (defined to be the fraction of neurons that show non-zero activity, averaged over stimuli). Lines are analytical estimations from §7.3; bars are simulation results. Vertical dotted lines are the optimal hidden layer size predicted by theory. We set $L_x = 50$ and $N = 30000$, as in Fig. 3. **B)** Optimal hidden layer size under sparse coding ($\rho_{CL} = 0.05$), with $N = 1.4L_x^{1.9}$, where we recover the 3/2 scaling. In both panels, we used the non-sparse ReLU ($\rho_{CL} = 0.5$) for the activation function of the teacher network, and set the teacher noise to $\sigma_t^2 = 0.1$.

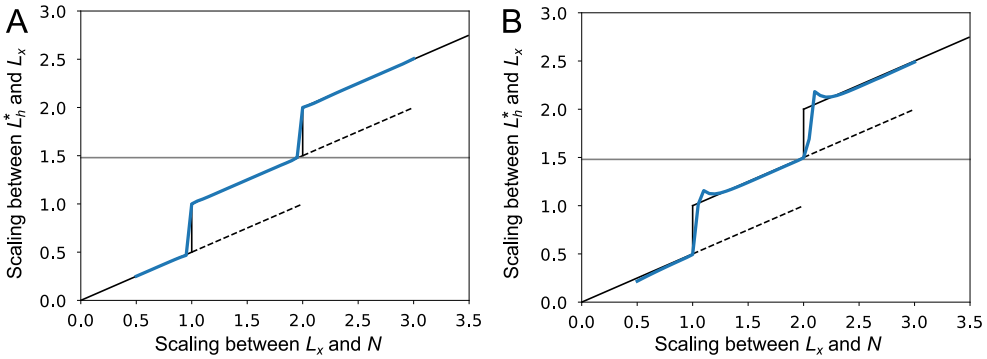

Figure S7: Scaling between the hidden layer size and the input layer size, versus scaling between the input layer size and the training data size. Analytical estimation of the relationship between the L_x-N scaling (the exponent γ of $N \propto L_x^{\gamma}$) and the L_h^*-L_x scaling (the exponent β of $L_h^* \propto L_x^{\beta}$) **A**) MLE; **B**) SGD. In both panels, the black lines are the analytical solution in the large L_x limit (see SI §5 for details); they are the same as the lines in Fig. 4F. The blues lines (computed for the range $\gamma = 0.5 - 3.0$) were estimated from the solution of Eq. (63) (panel A) and Eq. (113) (panel B), respectively. The exponents on the y-axes were estimated by performing linear regression on $(\log L_x, \log L_h^*)$ for $L_x = 10^8 - 10^{16}$.

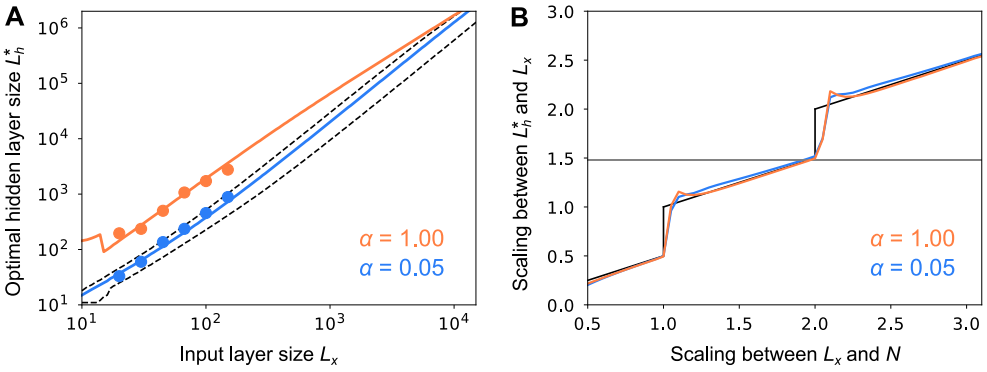

Figure S8: Scaling when learning under SGD is terminated after αN samples, but the circuit is optimized for the cumulative error over all N samples. This corresponds to the scenario where learning stops after sexual maturation. With this interpretation, N is the number of labeled samples an animal observes during its lifetime, and α is the fraction of the animal's lifetime it takes to reach sexual maturation. Here the total error is $\epsilon_\alpha = \alpha\epsilon_{cg}^{(\alpha N)} + (1-\alpha)\epsilon_{gen}^{(\alpha N)}$, where $\epsilon_{cg}^{(\alpha N)}$ is the cumulative generalization error from $n = 1, ..., \alpha N$ and $\epsilon_{gen}^{(\alpha N)}$ is the generalization error at $n = \alpha N$. **A**) Optimal hidden layer size with respect to ϵ_α as a function of L_x with $N = 19L_x^{1.96}$. The orange line is the same as the gray line in Fig. 5E. Black dotted lines are the analytical curves under $\alpha = 0.02$ (bottom) and $\alpha = 0.08$. **B)** Scaling between the hidden layer size and the input layer size versus scaling between the input layer size and the total amount of data. Lines are estimated from Eq. 131, not directly from simulations. The black line is the analytical solution in the large L_x and $\alpha \to 1$ limit, and the orange line is the same as the blue line in Fig. S7B. In both panels we set $\sigma^2 = 0.1$ and used SGD with a fixed learning rate.

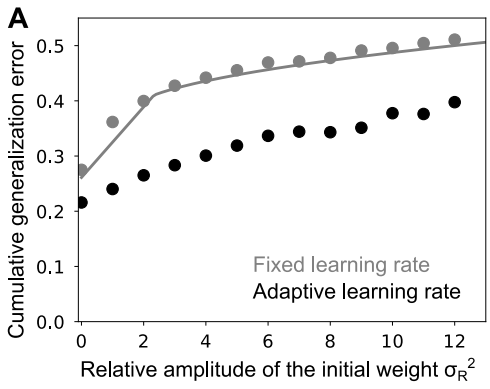

Figure S9: The cumulative generalization error ϵ_{cg}^{N} under various initial weight amplitude σ_R^2, when learning is performed with a fixed learning rate (gray), and an adaptive learning rate (black). The hidden layer size, L_h, was set to the optimal value estimated numerically in Fig. 5F. The initial projection weights were sampled from $\boldsymbol{w}_s^{(0)} \sim N(0, \sigma_R^2/L_h)$.

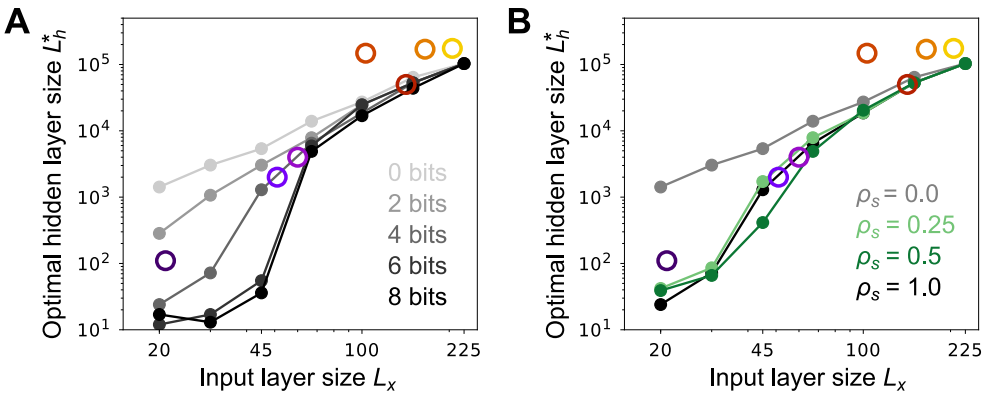

Figure S10: The optimal hidden layer size of the developmentally specified pathway. **A)** Optimal hidden layer size for a range of bits per synapse, s_b. Lines with 0, 2 and 4 bits correspond to Fig. 6C. **B)** A model with a sparse genetically specified pathway. Simulations were done with a range of connection probabilities, denoted ρ_s, which determines the probability that a connection from the input to the hidden layer of the genetically specified circuit is nonzero: $\rho_s \equiv \mathrm{Prob}[J_{ij}^p \neq 0]$). With this definition, $\rho_s = 0.0$ corresponds to a model without a genetic pathway, while $\rho_s = 1.0$ corresponds to a model with a fully connected genetic pathway. Gray and black lines ($\rho_s = 0$ and 1) correspond to Fig. 6C. See §6.2 for details.

1 Data analysis

1.1 Scaling in the invertebrate olfactory circuitry

Table 1 gives the number of glomeruli and Kenyon cells that were used to make Fig. 1B, along with the sources. All numbers are estimates for one hemisphere. It should be noted that the data is not well controlled. For instance, in some cases only the total number of mushroom body neurons are available, instead of the number of Kenyon cells, and different experimental techniques were used for different animals. Moreover, for locusts we used the number of olfactory receptor genes as an estimate of the effective number of glomeruli, because they have a unique micro-glomeruli structure which makes a direct comparison difficult [4]. There is usually a one-to-one correspondence between the number of olfactory receptor genes and the number of glomeruli in the invertebrate olfactory system, so this should be a good proxy of the effective number of glomeruli [5, 6]. For the total number of glomeruli, a comprehensive review is available [7]. Further explanation of the data for each species is given below.

Species	#Glomeruli	#KC	ref(G)	ref(KC)
Drosophila Melanogaster (larva)	21	110	[8, 9]	[10]
Drosophila Melanogaster (adult)	51	2000	[11]	[12]
Moth (Spodoptera litoralis)	60	4000	[13]	[14]
Locust	142*	50000	[15]	[16]
Apis mellifica (drone)	103	148000	[17]	[18]
Apis mellifica (worker)	165	170000	[17]	[18]
Cockroach (Periplaneta americana)	205	175000	[19]	[20]

Table 1. Number of glomeruli and Kenyon cells (KCs) for seven invertebrate species. *For locusts, the number of olfactory receptor genes is shown.

Drosophila Melanogaster (larvae)

Recent studies suggest that fruit fly larvae have a well functioning olfactory system, though the corresponding circuit is much smaller than that of adults [21]: there are only 21 glomeruli [8, 9], and about 110 Kenyon cells [10].

Drosophila Melanogaster (adults)

The adult fruit fly is the best studied species of insect. Their olfactory system contains 51 glomeruli that project to the antennal lobe [11], and around 2200 mushroom body neurons, of which about 2000 are Kenyon cells [12].

Moth (Spodoptera litoralis)

Several species of moth have been studied, and they all have around sixty glomeruli [7], including Spodoptera litoralis [13]. Less is known about the number of Kenyon cells, but one study found that the mushroom body of Spodoptera litoralis contains around 4000 of them [14].

Locust

Unlike most invertebrates, the locust olfactory circuit has a micro-glomeruli structure [4], meaning each micro-glomerulus receives input from multiple types of olfactory receptor neurons. This makes it difficult to compare with other species. However, we can use the number of olfactory receptor genes as a proxy, as discussed above. Under this assumption, the number is about 142 [15]. The number of Kenyon cells is estimated to be around 50000 [16].

Apis mellifica (drone)

Due to the caste system of honey bees, male honey bees (drones) do not engage in foraging or colony protection. Correspondingly, they have a smaller number of glomeruli compared to worker bees (103 vs 165 [17]), despite their larger body size. Their mushroom body is estimated to contain about 148000 neurons [18]. As shown in Fig. 1B, the drone bee is a clear outlier from the scaling law. This may be because of its unique ecological niche.

Apis mellifica (worker)

As mentioned above, worker bees have around 165 glomeruli [17], and the number of neurons in the mushroom body is around 170000 [18]. Note that the mushroom body of the worker honey bee is known to take part in visual navigation as well as olfaction [22].

Cockroach (Periplaneta americana)

Cockroaches are known to have excellent olfactory discrimination and learning ability [23]. They have about 205 glomeruli [19] and around 175000 Kenyon cells [20].

1.2 Correlation between the number of glomeruli and the duration of learning

The maximum longevity and average time from weaning to sexual maturation can be estimated from the AnAge database [24]. These are summarized in Table 2 for the six mammalian species used in our analysis (Fig. 1A) [1]. For species with different sexual maturation times for males and females, we took the average.

Species	common name	longevity (y)	weaning (d)	sexual maturity (d)	Δmaturation (d)
Mus musculus	mice	4.0	22	42	20
Rattus norvegicus	rats	3.8	25	80	55
Monodelphis domestica	opossums	5.1	53	122	69
Cavia porcellus	guinea pig	12	18	71	53
Mustela putorius	ferrets	11.1	63	317	254
Felis catus	cats	30	56	289	233

Table 2. Maximum longevity, the time of weaning and of sexual maturation, and the difference of the latter two, denoted Δmaturation, for the six mammalian species shown in Fig. 1A. y = year; d = day.

Figures S4A and S4B were obtained by plotting the data above (Table 2) against the number of glomeruli in Table 3 of [1].

2 Model setting

We consider a three layer student-teacher model. We assume that the generative model of the environment (the teacher model) is

$$y = \boldsymbol{w}_t \cdot g_t(\boldsymbol{J}_t \boldsymbol{x}) + \sigma_t \xi, \tag{1}$$

where $g_t(\cdot)$ is a pointwise nonlinearity, $\boldsymbol{x} \in \Re^{L_x}$ is the olfactory input as, $y \in \Re$ is the associated reward/valence/label, $\boldsymbol{w}_t \in \Re^{L_t}$ and $\boldsymbol{J}_t \in \Re^{L_t \times L_x}$ are random matrices with elements drawn from a zero mean Gaussian,

$$w_j^t \sim \mathcal{N}(0, 1/L_t) \tag{2a}$$

$$J_{ij}^t \sim \mathcal{N}(0, 1/L_x), \tag{2b}$$

and $\sigma_t \xi$ is the teacher noise, which reflects the probabilistic correspondence between $\boldsymbol{x}$ and y. Here ξ is a zero mean, unit variance Gaussian random variable. Throughout the text we use bold capital letters to denote matrices and bold small letter for vectors. As in the main text, vectors are defined as column vectors, and a superscript T denotes transpose (indicating a row vector). For readibility, we use a dot product to denote the inner product between two vectors.

The olfactory circuit (the student model) needs to mimic the teacher model to predict the reward/valence/label, y, given olfactory input, $\boldsymbol{x}$. We approximate this circuit by a three layer feedforward network,

$$\hat{y} = \boldsymbol{w}_s \cdot g_s(\boldsymbol{J}_s \boldsymbol{x}), \tag{3}$$

where, like $g_t(\cdot)$, $g_s(\cdot)$ is a pointwise nonlinearity, $\boldsymbol{w}_s \in \Re^{L_h}$ and $\boldsymbol{J}_s \in \Re^{L_h \times L_x}$. We assume that $\boldsymbol{J}_s$ is fixed and random with, as for the teacher network, entries drawn from a zero mean Gaussian,

$$J_{ij}^s \sim \mathcal{N}(0, 1/L_x)\,. \tag{4}$$

The readout weights, on the other hand, evolve with learning. Under SGD, their initial values are assumed to be a zero mean Gaussian, scaled by the parameter σ_R^2,

$$w_{si}^{(0)} \sim \mathcal{N}(0, \sigma_R^2/L_h) \tag{5}$$

where the superscript 0 denotes values before learning starts. The goal of learning is to tune the projection weights, $\boldsymbol{w}_s$, based on the samples generated from the teacher model, $D_N = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$. For analytical tractability we assume that the olfactory inputs, $\boldsymbol{x}$, are sampled from an independent Gaussian distribution

$$\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I})\,. \tag{6}$$

For the objective function, we use the mean squared error averaged over the input distribution, $p(\boldsymbol{x})$, and the teacher noise distribution, $p(\xi)$,

$$\epsilon_{gen} \equiv \left\langle \left[\boldsymbol{w}_t \cdot g_t(\boldsymbol{J}_t \boldsymbol{x}) + \sigma_t \xi - \boldsymbol{w}_s \cdot g_s(\boldsymbol{J}_s \boldsymbol{x}) \right]^2 \right\rangle_{p(\boldsymbol{x},\xi)} . \tag{7}$$

Under this loss function, the optimal projection weight, $\boldsymbol{w}_s^*$, is given by the standard expression for linear regression,

$$\boldsymbol{w}_s^* \equiv \left\langle g_s(\boldsymbol{J}_s \boldsymbol{x}) g_s(\boldsymbol{J}_s \boldsymbol{x})^T \right\rangle^{-1} \left\langle g_s(\boldsymbol{J}_s \boldsymbol{x}) g_t(\boldsymbol{J}_t \boldsymbol{x})^T \right\rangle \boldsymbol{w}_t . \tag{8}$$

When learning is unbiased, the generalization error divides cleanly into an approximation error and an estimation error,

$$\epsilon_{gen} = \sigma_t^2 + \left\langle \left(\left[\boldsymbol{w}_t \cdot g_t(\boldsymbol{J}_t \boldsymbol{x}) - \boldsymbol{w}_s^* \cdot g_s(\boldsymbol{J}_s \boldsymbol{x}) \right] + \left[(\boldsymbol{w}_s^* - \boldsymbol{w}_s) \cdot g_s(\boldsymbol{J}_s \boldsymbol{x}) \right] \right)^2 \right\rangle_{p(\boldsymbol{x},\xi)} = \sigma_t^2 + \epsilon_{apr} + \epsilon_{est}, \tag{9}$$

where

$$\epsilon_{apr} \equiv \left\langle \left[\boldsymbol{w}_t \cdot g_t(\boldsymbol{J}_t \boldsymbol{x}) - \boldsymbol{w}_s^* \cdot g_s(\boldsymbol{J}_s \boldsymbol{x}) \right]^2 \right\rangle_{p(\boldsymbol{x})} \tag{10a}$$

$$\epsilon_{est} \equiv \left\langle \left[(\boldsymbol{w}_s - \boldsymbol{w}_s^*) \cdot g_s(\boldsymbol{J}_s \boldsymbol{x}) \right]^2 \right\rangle_{p(\boldsymbol{x})} . \tag{10b}$$

The approximation error, ϵ_{apr}, depends only on the architecture, while the estimation error, ϵ_{est}, depends on both the choice of the learning method and the number of trials, N. Below we derive approximate analytical expressions for both ϵ_{apr} and ϵ_{est}, starting with the former.

3 Approximation error

To reduce clutter, we make the definitions

$$\boldsymbol{g}_t \equiv g_t(\boldsymbol{J}_t \boldsymbol{x}) \tag{11a}$$

$$\boldsymbol{g}_s \equiv g_s(\boldsymbol{J}_s \boldsymbol{x}) . \tag{11b}$$

In terms of these quantities, the approximation error is

$$\epsilon_{apr} = \boldsymbol{w}_t^T \left(\langle \boldsymbol{g}_t \boldsymbol{g}_t^T \rangle - \langle \boldsymbol{g}_t \boldsymbol{g}_s^T \rangle \boldsymbol{G}_s^{-1} \langle \boldsymbol{g}_s \boldsymbol{g}_t^T \rangle \right) \boldsymbol{w}_t \tag{12}$$

where the angle brackets represent an average over $p(\boldsymbol{x})$, the distribution of the input, and $\boldsymbol{G}_s$ is the uncentered hidden layer covariance matrix,

$$\boldsymbol{G}_s \equiv \langle \boldsymbol{g}_s \boldsymbol{g}_s^T \rangle . \tag{13}$$

Computing ϵ_{apr} is hard because it involves the inverse of the covariance matrix, $\boldsymbol{G}_s$. However, for the model we consider, the off-diagonal elements can be expanded in powers of $1/L_x^{1/2}$ (as we show below). We make use of this expansion to compute (approximately) the eigenvalues and eigenvectors of $\boldsymbol{G}_s$, and use those to find the inverse. That calculation is described next. In the bulk of the analysis we consider arbitrary nonlinear functions $g_s(\cdot)$ and $g_t(\cdot)$. In our numerical analysis we use ReLU and logistic functions.

Hidden layer covariance

The hidden layer covariance is computed by averaging over $\boldsymbol{x}$. Note, though, that wherever $\boldsymbol{x}$ appears it is multiplied by $\boldsymbol{J}_s$, so instead of averaging over $\boldsymbol{x}$ we can average over $\boldsymbol{u} \equiv \boldsymbol{J}_s \boldsymbol{x}$. Because $\boldsymbol{x}$ is Gaussian and white, $\boldsymbol{u}$ is also Gaussian, but it is correlated,

$$\boldsymbol{u} \sim N(0, \boldsymbol{J}_s \boldsymbol{J}_s^T) . \tag{14}$$

We thus have

$$(\boldsymbol{G}_s)_{ij} = \int du_i du_j \, p(u_i, u_j) g_s(u_i) g_s(u_j) \tag{15}$$

where $p(u_i, u_j)$ is a correlated Gaussian distribution with variance σ_i^2 and correlation coefficient ρ_{ij}; these quantities are give by

$$\sigma_i^2 = (\boldsymbol{J}_s \boldsymbol{J}_s^T)_{ii} \approx \left\langle (\boldsymbol{J}_s \boldsymbol{J}_s^T)_{ii} \right\rangle_{p(\boldsymbol{J}_s)} = 1. \tag{16a}$$

$$\rho_{ij} = \frac{(\boldsymbol{J}_s \boldsymbol{J}_s^T)_{ij}}{\sigma_i \sigma_j} \approx (\boldsymbol{J}_s \boldsymbol{J}_s^T)_{ij}\,,\ i \neq j\,. \tag{16b}$$

We will use $\sigma_i^2 = 1$ in what follows.

Let us first consider the diagonal terms which, under the approximation that $\sigma_i^2 = 1$, are all the same,

$$\langle g_s(u_i)^2 \rangle \approx \int_{-\infty}^{\infty} \frac{du_i}{\sqrt{2\pi}} \exp\left(-\frac{u_i^2}{2}\right) g_s(u_i)^2 \equiv D_0^s. \tag{17}$$

If $g_s(\cdot)$ is ReLU, we can compute D_0^s analytically, and it's given by $D_0^s = 1/2$; for other functions, D_0^s must be evaluated numerically (see §7.2). For the off-diagonal terms ($i \neq j$), again under the approximation $\sigma_i^2 = 1$, we have

$$\langle g_s(u_i) g_s(u_j) \rangle \approx \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{du_i du_j}{2\pi(1-\rho_{ij}^2)^{1/2}} \exp\left(-\frac{u_i^2 + u_j^2 - 2\rho_{ij} u_i u_j}{2(1-\rho_{ij}^2)}\right) g_s(u_i) g_s(u_j)\,. \tag{18}$$

Expansion of the ρ-dependent terms in Eq. (18) around $\rho_{ij} = 0$ gives

$$\frac{1}{\sqrt{1-\rho_{ij}^2}} \exp\left(-\frac{u_i^2 + u_j^2 - 2\rho_{ij} u_i u_j}{2(1-\rho_{ij}^2)}\right) = \exp\left(-\frac{u_i^2}{2} - \frac{u_j^2}{2}\right)\left[1 + u_i u_j \rho_{ij} + \frac{(1-u_i^2)(1-u_j^2)}{2}\rho_{ij}^2 + O(\rho_{ij}^3)\right]. \tag{19}$$

Consequently, when $i \neq j$, $\langle g_s(u_i) g_s(u_j) \rangle$ can be approximated as

$$\langle g_s(u_i) g_s(u_j) \rangle \approx C_0^{ss} + C_1^{ss} \rho_{ij} + C_2^{ss} \rho_{ij}^2, \tag{20}$$

where

$$C_0^{ss} \equiv \langle g_s(u) \rangle_{\mathcal{N}}^2 \tag{21a}$$

$$C_1^{ss} \equiv \langle u g_s(u) \rangle_{\mathcal{N}}^2 \tag{21b}$$

$$C_2^{ss} \equiv \frac{1}{2} \left\langle [1-u^2] g_s(u) \right\rangle_{\mathcal{N}}^2\,. \tag{21c}$$

The subscript $\mathcal{N}$ indicates an average over a standard Normal: for any function $g(u)$,

$$\langle g(u) \rangle_{\mathcal{N}} \equiv \int_{-\infty}^{\infty} \frac{du\, e^{-u^2/2}}{(2\pi)^{1/2}}\, g(u)\,. \tag{22}$$

To estimate the size of ρ_{ij}, we note that its mean and variance are given by, for $i \neq j$,

$$\left\langle \rho_{ij} \right\rangle_{p(\boldsymbol{J}_s)} = \left\langle [\boldsymbol{J}_s \boldsymbol{J}_s^T]_{ij} \right\rangle_{p(\boldsymbol{J}_s)} = 0, \tag{23a}$$

$$\left\langle \rho_{ij}^2 \right\rangle_{p(\boldsymbol{J}_s)} = \left\langle \left([\boldsymbol{J}_s \boldsymbol{J}_s^T]_{ij}\right)^2 \right\rangle_{p(\boldsymbol{J}_s)} = \frac{1}{L_x}. \tag{23b}$$

Consequently, the correlation, ρ_{ij}, is in the order of $1/L_x^{1/2}$. However, there are L_h times more off-diagonal terms than diagonal terms in the matrix $\boldsymbol{G}_s$, so we cannot ignore the ρ_{ij}-dependent terms in Eq. (20) unless $L_h^2 \ll L_x$. Nevertheless, for $L_x \gg 1$ (the relevant limit in our analysis), the correlation satisfies $|\rho_{ij}| \ll 1$, suggesting that for large L_x, a second-order Taylor expansion in ρ_{ij} should provide a good approximation to $\langle g(u_i) g(u_j) \rangle$. That's the approach we take here.

Combining the diagonal (Eq. (17)) and off-diagonal (Eq. (20)) terms, we can write the full covariance matrix as

$$\begin{aligned} \langle g_s(u_i) g_s(u_j) \rangle &\approx D_0^s \delta_{ij} + (C_0^{ss} + C_1^{ss} \rho_{ij} + C_2^{ss} \rho_{ij}^2)(1 - \delta_{ij}) \\ &= \left(D_0^s - (C_0^{ss} + C_1^{ss} \rho_{ij} + C_2^{ss} \rho_{ij}^2)\right)\delta_{ij} + C_0^{ss} + C_1^{ss} \rho_{ij} + C_2^{ss} \rho_{ij}^2\,. \end{aligned} \tag{24}$$

Strictly speaking, ρ_{ij} is not defined at $i = j$, but we can choose it arbitrarily without changing the covariance matrix. We thus extend Eq. (16b) to include $i = j$, and write (again using $\sigma_i = 1$)

$$\rho_{ij} = (\boldsymbol{J}_s \boldsymbol{J}_s^T)_{ij}\,, \tag{25}$$

now valid for $i = j$ as well as $i \neq j$ (and with the convention that $\rho_{ii} = 1$). With this definition, Eq. (24) becomes

$$\langle g_s(u_i) g_s(u_j) \rangle \approx \delta_s \delta_{ij} + (C_0^{ss} + C_2^{ss} \langle \rho^2 \rangle) + C_1^{ss} \rho_{ij} + C_2^{ss} (\rho_{ij}^2 - \langle \rho^2 \rangle) \tag{26}$$

where $\langle \rho^2 \rangle$ is defined in Eq. (23b) and

$$\delta_s \equiv D_0^s - (C_0^{ss} + C_1^{ss} + C_2^{ss}) \,. \tag{27}$$

It is straightforward to show that

$$\rho_{ij}^2 - \langle \rho^2 \rangle = \sum_{m=1}^{L_x} \sum_{l=m+1}^{L_x} M_{i,[m,l]}^s M_{j,[m,l]}^s \tag{28}$$

where $[m, l]$ is a compositional index, and $\boldsymbol{M}_s$ is an $L_h \times L_x(L_x - 1)/2$ matrix,

$$M_{i,[m,l]}^s \equiv \sqrt{2} J_{im}^s J_{il}^s \,. \tag{29}$$

Combining these expressions, and using the fact that $\langle \rho^2 \rangle = 1/L_x$ (Eq. (23b)), which is small compared to 1, the covariance matrix simplifies to

$$\boldsymbol{G}_s \approx \delta_s \boldsymbol{I} + C_0^{ss} \mathbf{1}_h \mathbf{1}_h^T + C_1^{ss} \boldsymbol{J}_s \boldsymbol{J}_s^T + C_2^{ss} \boldsymbol{M}_s \boldsymbol{M}_s^T \tag{30}$$

where $\mathbf{1}_h \in \Re^{L_h}$ is a vector in which all the elements are one. The first two matrices in this expression are a scaled identity matrix and a matrix with the same value everywhere. The third, $\boldsymbol{J}_s \boldsymbol{J}_s^T$, is a Wishart matrix, so its eigenspectrum follows a Marchenko-Pastur distribution [25]; from Eq. (4) we see that the parameters of that distribution are $(\sigma^2, \overline{\lambda}) = (1, L_h/L_x)$. Similarly, although the columns of $\boldsymbol{M}_s$ are not independent, given that they have zero correlation we assume that the eigenspectrum of $\boldsymbol{M}_s \boldsymbol{M}_s^T$ also follows a Marchenko-Pastur distribution; from Eq. (29) we see that the parameters are $(\sigma^2, \overline{\lambda}) = (1, 2L_h/L_x^2)$.

Essentially identical analysis gives us the covariance between the hidden units of the teacher and student networks,

$$\langle \boldsymbol{g}_t \boldsymbol{g}_s^T \rangle \approx C_0^{ts} \mathbf{1}_t \mathbf{1}_h^T + C_1^{ts} \boldsymbol{J}_t \boldsymbol{J}_s^T + C_2^{ts} \boldsymbol{M}_t \boldsymbol{M}_s^T \,, \tag{31}$$

where $\boldsymbol{M}_t$ is an $L_t \times L_x(L_x - 1)/2$ matrix analogous to $\boldsymbol{M}_s$,

$$M_{i,[k,l]}^t \equiv \sqrt{2} J_{ik}^t J_{il}^s, \tag{32}$$

and $C_0^{ts}, C_1^{ts}, C_2^{ts}$ are natural extensions of $C_0^{ss}, C_1^{ss}, C_2^{ss}$ (Eq. (21)), the difference being that one of the student averages becomes a teacher average,

$$C_0^{ts} \equiv \langle g_t(u) \rangle_{\mathcal{N}} \langle g_s(u) \rangle_{\mathcal{N}} \tag{33a}$$

$$C_1^{ts} \equiv \langle u g_t(u) \rangle_{\mathcal{N}} \langle u g_s(u) \rangle_{\mathcal{N}} \tag{33b}$$

$$C_2^{ts} \equiv \frac{1}{2} \left\langle \left(1 - u^2\right) g_t(u) \right\rangle_{\mathcal{N}} \left\langle \left(1 - u^2\right) g_s(u) \right\rangle_{\mathcal{N}} \,. \tag{33c}$$

To calculate the approximation error, ϵ_{apr} (Eq. (12)), we need $\boldsymbol{G}_s^{-1}$. For that we express $\boldsymbol{G}_s$ in terms of its eigenvalues and eigenvectors, from which the inverse follows easily. That analysis, which is nontrivial, is carried out in §8; we simply report the result here,

$$\boldsymbol{G}_s \approx \lambda^{(0)} \boldsymbol{v}^{(0)} \left(\boldsymbol{v}^{(0)}\right)^T + \sum_{k=1}^{L_1} \lambda_k^{(1)} \boldsymbol{v}_k^{(1)} \left(\boldsymbol{v}_k^{(1)}\right)^T + \sum_{k=1}^{L_2} \lambda_k^{(2)} \boldsymbol{v}_k^{(2)} \left(\boldsymbol{v}_k^{(2)}\right)^T + \sum_{k=1}^{L_r} \lambda^{(r)} \boldsymbol{v}_k^{(r)} \left(\boldsymbol{v}_k^{(r)}\right)^T \,. \tag{34}$$

As shown in §8, the eigenvalues are

$$\lambda^{(0)} = C_0^{ss} (c_0 + L_h) \tag{35a}$$

$$\lambda_k^{(1)} = C_1^{ss} \left(c_1 + \tilde{\lambda}_k^{(1)}\right) \tag{35b}$$

$$\lambda_k^{(2)} = C_2^{ss} \left(c_2 + \tilde{\lambda}_k^{(2)}\right) \tag{35c}$$

$$\lambda^{(r)} = \delta_s \,, \tag{35d}$$

where $\tilde{\lambda}_k^{(1)}$ and $\tilde{\lambda}_k^{(2)}$ are eigenvalues of $\boldsymbol{J}_s\boldsymbol{J}_s^T$ and $\boldsymbol{M}_s\boldsymbol{M}_s^T$, respectively, and the coefficients are

$$c_0 \equiv \frac{\delta_s + C_1^{ss} + C_2^{ss}}{C_0^{ss}} \tag{36a}$$

$$c_1 \equiv \frac{\delta_s + C_2^{ss}}{C_1^{ss}} \tag{36b}$$

$$c_2 \equiv \frac{\delta_s}{C_2^{ss}}\,. \tag{36c}$$

The rank of $\boldsymbol{J}_s\boldsymbol{J}_s^T$ and $\boldsymbol{M}_s\boldsymbol{M}_s^T$ are L_1 and L_2, respectively, and L_r is such that it picks up any dimensionality uncaptured by the second-order expansion (because $\boldsymbol{G}_s$ is typically full rank under a nonlinear activation function),

$$L_1 = \min[L_x, L_h - 1] \approx \min[L_x, L_h] \tag{37a}$$

$$L_2 = \min\left[\frac{1}{2}L_x(L_x-1), L_h - L_x - 1\right]^+ \approx \min\left[\frac{L_x^2}{2}, L_h - L_x\right]^+ \tag{37b}$$

$$L_r = \left[L_h - \left(1 + L_x + \frac{1}{2}L_x(L_x-1)\right)\right]^+ \approx \left[L_h - \frac{L_x^2}{2}\right]^+ \tag{37c}$$

where the superscript $+$ indicates the threshold-linear operation: $[x]^+ = x$ if $x > 0$ and 0 otherwise, and the approximations are valid because we are interested in the large L_x and L_h limit. The first of these two quantities, L_1 and L_2, are plotted in Fig. 5A.

We are now in a position to derive an explicit expression for the approximation error, ϵ_{apr}, given in Eq. (12). Noticing that $\boldsymbol{w}_t$ is a zero-mean random vector (Eq. (2a)), in the large L_t limit we may make the approximation $\boldsymbol{w}_t\boldsymbol{w}_t^T \approx \boldsymbol{I}/L_t$. Consequently,

$$\epsilon_{apr} \approx \frac{1}{L_t}\,\mathsf{Tr}\left[\langle \boldsymbol{g}_t\boldsymbol{g}_t^T\rangle - \langle \boldsymbol{g}_t\boldsymbol{g}_s^T\rangle \boldsymbol{G}_s^{-1}\langle \boldsymbol{g}_s\boldsymbol{g}_t^T\rangle\right]\,. \tag{38}$$

The first term is given by

$$\frac{1}{L_t}\,\mathsf{Tr}\left[\langle \boldsymbol{g}_t\boldsymbol{g}_t^T\rangle\right] = \int_{-\infty}^{\infty}\frac{du}{\sqrt{2\pi}}\exp\left(-\frac{u^2}{2}\right) g_t(u)^2 \equiv D_0^t\,. \tag{39}$$

To compute the second term, we start by writing it

$$\frac{1}{L_t}\,\mathsf{Tr}\left[\langle \boldsymbol{g}_t\boldsymbol{g}_s^T\rangle \boldsymbol{G}_s^{-1}\langle \boldsymbol{g}_s\boldsymbol{g}_t^T\rangle\right] = \frac{1}{L_t}\,\mathsf{Tr}\left[\langle \boldsymbol{g}_s\boldsymbol{g}_t^T\rangle\langle \boldsymbol{g}_t\boldsymbol{g}_s^T\rangle \boldsymbol{G}_s^{-1}\right]\,. \tag{40}$$

Using Eq. (31), we have

$$\begin{aligned}\frac{1}{L_t}\langle \boldsymbol{g}_s\boldsymbol{g}_t^T\rangle\langle \boldsymbol{g}_t\boldsymbol{g}_s^T\rangle &\approx \frac{1}{L_t}\left[C_0^{ts}\mathbf{1}_h\mathbf{1}_t^T + C_1^{ts}\boldsymbol{J}_s\boldsymbol{J}_t^T + C_2^{ts}\boldsymbol{M}_s\boldsymbol{M}_t^T\right]\left[C_0^{ts}\mathbf{1}_t\mathbf{1}_h^T + C_1^{ts}\boldsymbol{J}_t\boldsymbol{J}_s^T + C_2^{ts}\boldsymbol{M}_t\boldsymbol{M}_s^T\right]\\ &\approx (C_0^{ts})^2\mathbf{1}_h\mathbf{1}_h^T + \frac{(C_1^{ts})^2\boldsymbol{J}_s\boldsymbol{J}_s^T}{L_x} + \frac{(C_2^{ts})^2\boldsymbol{M}_s\boldsymbol{M}_s^T}{L_x^2/2}\,.\end{aligned} \tag{41}$$

The second line follows because the cross terms, $\mathbf{1}_h^T\boldsymbol{J}_t$, $\mathbf{1}_h^T\boldsymbol{M}_t$ and $\boldsymbol{J}_t^T\boldsymbol{M}_t$, are all approximately zero, so long as L_t is sufficiently large. To derive this expression we took the large L_x limit, and replaced $L_x(L_x-1)$ with L_x^2. Combining this with the expression for $\boldsymbol{G}_s$, Eq. (34), from which it is easy to write down the inverse, and making use of Eqs. (35) and (173), we arrive at

$$\frac{1}{L_t}\,\mathsf{Tr}\left[\langle \boldsymbol{g}_s\boldsymbol{g}_t^T\rangle\langle \boldsymbol{g}_t\boldsymbol{g}_s^T\rangle \boldsymbol{G}_s^{-1}\right] = \frac{(C_0^{ts})^2}{C_0^{ss}}\frac{L_h}{c_0 + L_h} + \frac{1}{L_x}\frac{(C_1^{ts})^2}{C_1^{ss}}\sum_{k=1}^{L_1}\frac{\tilde{\lambda}_k^{(1)}}{c_1 + \tilde{\lambda}_k^{(1)}} + \frac{1}{L_x^2/2}\frac{(C_2^{ts})^2}{C_2^{ss}}\sum_{k=1}^{L_2}\frac{\tilde{\lambda}_k^{(2)}}{c_2 + \tilde{\lambda}_k^{(2)}}\,. \tag{42}$$

Given our assumption that the eigenvalue spectrum of both $\boldsymbol{J}_s\boldsymbol{J}_s^T$ and $\boldsymbol{M}_s\boldsymbol{M}_s^T$ follow the Marchenko-Pastur distribution, the sums over the eigenvalues turn into averages over the Marchenko-Pastur distribution. Those averages, which are tedious but straightforward, are computed in §9, and we arrive at

$$\begin{aligned}\frac{1}{L_t}\,\mathsf{Tr}\left[\langle \boldsymbol{g}_s\boldsymbol{g}_t^T\rangle\langle \boldsymbol{g}_t\boldsymbol{g}_s^T\rangle \boldsymbol{G}_s^{-1}\right] &= \frac{(C_0^{ts})^2}{C_0^{ss}}\left[1 - \frac{c_0}{c_0 + L_h}\right] + \frac{(C_1^{ts})^2}{C_1^{ss}}\left[1 - f\left(\frac{L_h}{L_x}; c_1\right)\right]\\ &\quad + \frac{(C_2^{ts})^2}{C_2^{ss}}\left[1 - \frac{L_x}{L_h}\right]^+\left[1 - f\left(\frac{L_h}{L_x^2/2}; \frac{c_2}{1 - L_x/L_h}\right)\right]\,,\end{aligned} \tag{43}$$

where $f(\bar{\lambda}; c)$ is defined in Eq. (176); we repeat its definition here for convenience,

$$f(\bar{\lambda}; c) \equiv \frac{\sqrt{\left(\bar{\lambda} - 1 + c\right)^2 + 4c} - \left(\bar{\lambda} - 1 + c\right)}{2} \,. \tag{44}$$

This function has relatively simple asymptotic behavior: it is 1 when $\bar{\lambda} = 0$ and falls off as $c/\bar{\lambda}$ when $\bar{\lambda} \gg c$.

Combining Eq. (43) with the first term in the expression for the approximation error, Eq. (39), and inserting that into Eq. (12), we arrive at

$$\epsilon_{apr} \approx \delta_{ts} + \sum_{q=0}^{2} \frac{(C_q^{ts})^2}{C_q^{ss}} f_q(L_h) \tag{45}$$

where

$$f_0(L_h) \equiv \frac{c_0}{c_0 + L_h} \tag{46a}$$

$$f_1(L_h) \equiv f\left(\frac{L_h}{L_x}; c_1\right) \tag{46b}$$

$$f_2(L_h) \equiv \min\left[1, \frac{L_x}{L_h}\right] + \left[1 - \frac{L_x}{L_h}\right]^+ f\left(\frac{L_h}{L_x^2/2}; \frac{c_2}{1 - L_x/L_h}\right) \approx f\left(\frac{L_h}{L_x^2/2}; c_2\right) \tag{46c}$$

and

$$\delta_{ts} \equiv D_0^t - \frac{(C_0^{ts})^2}{C_0^{ss}} - \frac{(C_1^{ts})^2}{C_1^{ss}} - \frac{(C_2^{ts})^2}{C_2^{ss}} \,. \tag{47}$$

The approximation made in Eq. (46c) is accurate everywhere except the region $L_x \lesssim L_h$; that's because $f(\overline{\lambda}; c) \to 1$ when $\overline{\lambda} \ll 1$. From Eqs. (45) and (46), we recover the expression for the approximation error in the main text, with coefficients given by

$$\alpha \equiv \delta_{ts} \tag{48a}$$

$$a_0 \equiv c_0 (C_0^{ts})^2 / C_0^{ss} \tag{48b}$$

$$a_q \equiv (C_q^{ts})^2 / C_q^{ss} \,, \tag{48c}$$

with $q = 1, 2$ in the last expression. For Eq. (48b) we used $c_0 \ll L_h$.

4 Estimation error

The estimation error, which is given Eq. (10b), can be written

$$\epsilon_{est} \equiv (\boldsymbol{w}_s^* - \boldsymbol{w}_s)^T \boldsymbol{G}_s (\boldsymbol{w}_s^* - \boldsymbol{w}_s). \tag{49}$$

This quantity is a random variable that depends on the data. We thus consider its mean, which is the expectation over the distribution of training data,

$$\bar{\epsilon}_{est} \equiv \left\langle (\boldsymbol{w}_s^* - \boldsymbol{w}_s)^T \boldsymbol{G}_s (\boldsymbol{w}_s^* - \boldsymbol{w}_s) \right\rangle_{p(\boldsymbol{x}_{1:N}, y_{1:N})} \tag{50}$$

where $\{\boldsymbol{x}_{1:N}, y_{1:N}\}$ is the training data. We first consider maximum likelihood, then stochastic gradient descent.

4.1 Estimation error under maximum likelihood (MLE) learning

As the teacher noise, $\sigma_t \xi$, is Gaussian, given N training points $D_N = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$ with $y_n = \boldsymbol{w}_t \cdot g(\boldsymbol{J}_t \boldsymbol{x}_n) + \sigma_t \xi_n$, the MLE weights of the student network are given by the usual expression for least squares minimization,

$$\boldsymbol{w}_s = \left(\frac{1}{N} \sum_{n=1}^{N} g(\boldsymbol{J}_s \boldsymbol{x}_n) g(\boldsymbol{J}_s \boldsymbol{x}_n)^T\right)^{-1} \left(\frac{1}{N} \sum_{n=1}^{N} g(\boldsymbol{J}_s \boldsymbol{x}_n) y_n\right). \tag{51}$$

Note that $\sum_{n=1}^{N} g(\boldsymbol{J}_s \boldsymbol{x}_n) g(\boldsymbol{J}_s \boldsymbol{x}_n)^T$ is not invertible unless $N > L_h$, so we work in that regime. Denoting

$$\boldsymbol{g}_t^n \equiv g(\boldsymbol{J}_t \boldsymbol{x}_n) \tag{52a}$$

$$\boldsymbol{g}_s^n \equiv g(\boldsymbol{J}_s \boldsymbol{x}_n) \tag{52b}$$

$$\boldsymbol{G}_s^{(N)} \equiv \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{g}_s^n \left(\boldsymbol{g}_s^n\right)^T \tag{52c}$$

and noting that $y_n = \boldsymbol{w}_t \cdot \boldsymbol{g}_t^n + \sigma_t \xi_n$, $\boldsymbol{w}_s - \boldsymbol{w}_s^*$ is written

$$\boldsymbol{w}_s - \boldsymbol{w}_s^* = \left(\boldsymbol{G}_s^{(N)}\right)^{-1} \left(\frac{1}{N} \sum_{n=1}^{N} \boldsymbol{g}_s^n \left[\boldsymbol{w}_t \cdot \boldsymbol{g}_t^n + \sigma_t \xi_n - \boldsymbol{w}_s^* \cdot \boldsymbol{g}_s^n \right] \right). \tag{53}$$

Inserting this into Eq. (49), we have

$$\begin{aligned} \bar{\epsilon}_{est} = \frac{1}{N^2} \Bigg\langle & \left(\sum_{n=1}^{N} \left[\boldsymbol{w}_t \cdot \boldsymbol{g}_t^n + \sigma_t \xi_n - \boldsymbol{w}_s^* \cdot \boldsymbol{g}_s^n \right] \left(\boldsymbol{g}_s^n\right)^T \right) \left(\boldsymbol{G}_s^{(N)}\right)^{-1} \boldsymbol{G}_s \left(\boldsymbol{G}_s^{(N)}\right)^{-1} \\ & \left(\sum_{n'=1}^{N} \boldsymbol{g}_s^{n'} \left[\boldsymbol{w}_t \cdot \boldsymbol{g}_t^{n'} + \sigma_t \xi_{n'} - \boldsymbol{w}_s^* \cdot \boldsymbol{g}_s^{n'} \right] \right) \Bigg\rangle . \end{aligned} \tag{54}$$

The first observation is that the n and n'-dependent terms are independent when $n \neq n'$. Consequently, the double sum over n and n' can be replaced by its diagonal elements,

$$\bar{\epsilon}_{est} \approx \frac{1}{N^2} \left\langle \sum_{n=1}^{N} \left[\boldsymbol{w}_t \cdot \boldsymbol{g}_t^n + \sigma_t \xi_n - \boldsymbol{w}_s^* \cdot \boldsymbol{g}_s^n \right]^2 \left(\boldsymbol{g}_s^n\right)^T \left(\boldsymbol{G}_s^{(N)}\right)^{-1} \boldsymbol{G}_s \left(\boldsymbol{G}_s^{(N)}\right)^{-1} \boldsymbol{g}_s^n \right\rangle . \tag{55}$$

Second, we assume that $\left[\boldsymbol{w}_t \cdot \boldsymbol{g}_t^n - \boldsymbol{w}_s^* \cdot \boldsymbol{g}_s^n\right]^2$ and $(\boldsymbol{g}_s^n)^T \boldsymbol{g}_s^n$ average independently. Using Eqs. (10a) and (52c), this leads to

$$\begin{aligned} \bar{\epsilon}_{est} &\approx \left\langle \left[\boldsymbol{w}_t \cdot \boldsymbol{g}_t + \sigma_t \xi - \boldsymbol{w}_s^* \cdot \boldsymbol{g}_s \right]^2 \right\rangle \frac{1}{N} \left\langle \mathsf{Tr} \left[\left(\boldsymbol{G}_s^{(N)}\right)^{-1} \boldsymbol{G}_s \left(\boldsymbol{G}_s^{(N)}\right)^{-1} \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{g}_s^n (\boldsymbol{g}_s^n)^T \right] \right\rangle \\ &= \left(\epsilon_{apr} + \sigma_t^2\right) \frac{1}{N} \mathsf{Tr} \left[\left(\boldsymbol{G}_s^{(N)}\right)^{-1} \boldsymbol{G}_s \right] . \end{aligned} \tag{56}$$

We'll compute the the trace term in the limit N and L_h go to infinity, with the ratio L_h/N fixed at some value less than 1. We start by turning $\boldsymbol{G}_s^{(N)}$ into a zero mean matrix. To that end, we let $\boldsymbol{G}_s^{(N)} \equiv \delta \boldsymbol{G}_s^{(N)} + \bar{\boldsymbol{g}}_N \bar{\boldsymbol{g}}_N^T$ where

$$\delta \boldsymbol{G}_s^{(N)} \equiv \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{g}_s^n - \bar{\boldsymbol{g}}_n)(\boldsymbol{g}_s^n - \bar{\boldsymbol{g}}_n)^T \tag{57}$$

and

$$\bar{\boldsymbol{g}}_N \equiv \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{g}_s^n \approx \langle \boldsymbol{g}_s \rangle_{p(\boldsymbol{x})} \equiv \bar{\boldsymbol{g}} . \tag{58}$$

Approximating $\boldsymbol{G}_s^{(N)}$ with $\delta \boldsymbol{G}_s^{(N)} + \bar{\boldsymbol{g}} \bar{\boldsymbol{g}}^T$, and applying the Sherman-Morrison formula, we have

$$\left(\boldsymbol{G}_s^{(N)}\right)^{-1} \approx \left(\delta \boldsymbol{G}_s^{(N)} + \bar{\boldsymbol{g}} \bar{\boldsymbol{g}}^T\right)^{-1} = \left(\delta \boldsymbol{G}_s^{(N)}\right)^{-1} - \frac{\left(\delta \boldsymbol{G}_s^{(N)}\right)^{-1} \bar{\boldsymbol{g}} \, \bar{\boldsymbol{g}}^T \left(\delta \boldsymbol{G}_s^{(N)}\right)^{-1}}{1 + \bar{\boldsymbol{g}}^T \left(\delta \boldsymbol{G}_s^{(N)}\right)^{-1} \bar{\boldsymbol{g}}} . \tag{59}$$

To compute our calculation, we need an approximation for $\boldsymbol{G}_s$. In §3, we used Eq. (30). Here we make a more severe approximation: decomposing $\boldsymbol{G}_s$ as $\delta \boldsymbol{G}_s + \bar{\boldsymbol{g}} \bar{\boldsymbol{g}}^T$, we use $\delta \boldsymbol{G}_s \approx \sigma_M^2 \boldsymbol{I}$ where $\sigma_M^2 = \langle (g_{s,i}^n - \bar{g}_{n,i})^2 \rangle$. This is consistent with approximating $\delta \boldsymbol{G}_s^{(N)}$ as a Wishart matrix $W_{L_h}(\frac{\sigma_M^2}{N} \boldsymbol{I}, N)$, which converges to $\sigma_M^2 \boldsymbol{I}$ as $N \to \infty$. Inserting this, along with the above expression for $\boldsymbol{G}_s^{(N)}$, into Eq. (56), we arrive, after a small amount of algebra, at

$$\mathsf{Tr} \left[\left(\boldsymbol{G}_s^{(N)}\right)^{-1} \boldsymbol{G}_s \right] \approx \sigma_M^2 \mathsf{Tr} \left[\left(\delta \boldsymbol{G}_s^{(N)}\right)^{-1} \right] + 1 - \frac{1 + \sigma_M^2 \bar{\boldsymbol{g}}^T \left(\delta \boldsymbol{G}_s^{(N)}\right)^{-2} \bar{\boldsymbol{g}}}{1 + \bar{\boldsymbol{g}}^T \left(\delta \boldsymbol{G}_s^{(N)}\right)^{-1} \bar{\boldsymbol{g}}} . \tag{60}$$

The first term is proportional to L_h, the dimensionality of $\delta \boldsymbol{G}_s^{(N)}$. The last is bounded by $(1+\sigma_M^2|\bar{\boldsymbol{g}}|^2/\lambda_{\min}^2)/(1+|\bar{\boldsymbol{g}}|^2/\lambda_{\max})$ where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimum and maximum eigenvalues of $\delta \boldsymbol{G}_s^{(N)}$. Since $\delta \boldsymbol{G}_s^{(N)}$ is the sum of N outer products of random vectors, its spectrum approximately follows a Marchenko-Pastur distribution with parameters $(\sigma^2, \overline{\lambda}) = (\sigma_M^2, L_h/N)$. Assuming L_h is not too close to N, the eigenvalues are $\mathcal{O}(1)$. Consequently, the above expression is dominated by the first term. Restoring the prefactor $1/N$, we have

$$\frac{1}{N}\mathsf{Tr}\left[\left(\boldsymbol{G}_s^{(N)}\right)^{-1}\boldsymbol{G}_s\right] \approx \frac{\sigma_M^2 L_h}{N}\left\langle\frac{1}{\lambda}\right\rangle_{MP(\sigma_M^2, L_h/N)} = \frac{L_h}{N-L_h} \tag{61}$$

where the average over λ^{-1} (which gives us the second equality) is computed in §9 (see in particular Eq. (182)).

Equation (61) is consistent with previous work on linear regression using the replica method [26]. Inserting Eq. (61) into (56), we arrive at

$$\bar{\epsilon}_{est} \approx (\epsilon_{apr} + \sigma_t^2)\frac{L_h}{N-L_h}\,. \tag{62}$$

Consequently, the generalization error, $\epsilon_{gen} = \bar{\epsilon}_{est} + \epsilon_{apr} + \sigma_t^2$, Eq. (9), is given approximately by

$$\epsilon_{gen} \approx (\epsilon_{apr} + \sigma_t^2)\,\frac{N}{N-L_h}\,. \tag{63}$$

4.2 Estimation error under stochastic gradient descent (SGD) learning

In an online setting, it is more realistic to consider stochastic gradient descent rather than maximum likelihood, the former given by

$$\boldsymbol{w}_s^{(n)} = \boldsymbol{w}_s^{(n-1)} + \eta(y_n - \hat{y}_n)\boldsymbol{g}_s^n \tag{64}$$

where $\boldsymbol{g}_s^n$ is defined in Eq. (52b) and $\hat{y}_n = \boldsymbol{w}_s^{(n-1)}\cdot\boldsymbol{g}_s^n$ (Eq. (3)). Making the definition

$$\boldsymbol{u}_n \equiv \boldsymbol{w}_s^{(n)} - \boldsymbol{w}_s^*, \tag{65}$$

the update rule for $\boldsymbol{u}_n$ is

$$\begin{aligned}\boldsymbol{u}_n &= \left[\boldsymbol{I} - \eta\boldsymbol{g}_s^n(\boldsymbol{g}_s^n)^T\right]\boldsymbol{u}_{n-1} + \eta\left[\boldsymbol{w}_t\cdot\boldsymbol{g}_t^n - \boldsymbol{w}_s^*\cdot\boldsymbol{g}_s^n + \sigma_t\xi_n\right]\boldsymbol{g}_s^n \\ &= \left[\boldsymbol{I} - \eta\boldsymbol{G}_s\right]\boldsymbol{u}_{n-1} + \eta\left[\boldsymbol{G}_s - \boldsymbol{g}_s^n(\boldsymbol{g}_s^n)^T\right]\boldsymbol{u}_{n-1} + \eta\left[\boldsymbol{w}_t\cdot\boldsymbol{g}_t^n - \boldsymbol{w}_s^*\cdot\boldsymbol{g}_s^n + \sigma_t\xi_n\right]\boldsymbol{g}_s^n\end{aligned} \tag{66}$$

where, recall, $\boldsymbol{G}_s$ is the hidden layer covariance, defined in Eq. (13). After n updates, the estimation error (Eq. (10b)) is

$$\epsilon_{est}^{(n)} = \boldsymbol{u}_n^T\boldsymbol{G}_s\boldsymbol{u}_n. \tag{67}$$

It is convenient to work in a basis spanned by the eigenvectors of $\boldsymbol{G}_s$. That basis is given in Eq. (34), which shows a great deal of structure, and in particular a division into four components. Later we will use that structure, but for now we adopt a notation that hides it: we simply write $\boldsymbol{v}_\mu$ and λ_μ for the μ^{th} eigenvector and eigenvalue of $\boldsymbol{G}_s$ (i.e., $\boldsymbol{G}_s\boldsymbol{v}_\mu = \lambda_\mu\boldsymbol{v}_\mu$). We then make the change of variables

$$\boldsymbol{u}_n = \sum_\mu m_{\mu,n}\boldsymbol{v}_\mu\,. \tag{68}$$

In the new variables, the estimation error is

$$\epsilon_{est}^{(n)} = \sum_\mu \lambda_\mu m_{\mu,n}^2\,. \tag{69}$$

We'll first find the update rules for $m_{\mu,n}$, then use them to find the update rules for $m_{\mu,n}^2$. Taking the eigenvectors to be orthonormal, we have $m_{\mu,n} = \boldsymbol{v}_\mu\cdot\boldsymbol{u}_n$; applying this to Eq. (66) yields

$$m_{\mu,n} = (1-\eta\lambda_\mu)m_{\mu,n-1} + \eta\boldsymbol{v}_\mu^T\left[\boldsymbol{G}_s - \boldsymbol{g}_s^n(\boldsymbol{g}_s^n)^T\right]\boldsymbol{u}_{n-1} + \eta\left[\boldsymbol{w}_t\cdot\boldsymbol{g}_t^n - \boldsymbol{w}_s^*\cdot\boldsymbol{g}_s^n + \sigma_t\xi_n\right]\boldsymbol{v}_\mu\cdot\boldsymbol{g}_s^n\,. \tag{70}$$

Squaring both sides gives us an expression for $m_{\mu,n}^2$. To simplify that expression, we assume that the mean dynamics of $m_{\mu,n}^2$ is described by the dynamics of the mean, $\langle m_{\mu,n}^2\rangle$, where the average is over the distribution of the input $\boldsymbol{x}$ and the teacher noise ξ. To simplify notation, below we suppress the label n that appears on $\boldsymbol{g}_s^n, \boldsymbol{g}_t^n$, and ξ_n. (Note that $\boldsymbol{g}_s^n$ and $\boldsymbol{g}_t^n$ depend on n only through $\boldsymbol{x}_n$; see Eqs. (52a) and (52b)). The first term on the right hand side of Eq. (70) is

independent of $\boldsymbol{x}$, and the second two terms, which do depend on $\boldsymbol{x}$, are both zero mean. We assume those terms are uncorrelated, so we have

$$\begin{aligned} m_{\mu,n}^2 = (1-\eta\lambda_\mu)^2 m_{\mu,n-1}^2 &+ \eta^2 \left\langle \boldsymbol{u}_{n-1}^T [\boldsymbol{G}_s - \boldsymbol{g}_s \boldsymbol{g}_s^T] \boldsymbol{v}_\mu \boldsymbol{v}_\mu^T [\boldsymbol{G}_s - \boldsymbol{g}_s \boldsymbol{g}_s^T] \boldsymbol{u}_{n-1} \right\rangle \\ &+ \eta^2 \left\langle \left[\boldsymbol{w}_t \cdot \boldsymbol{g}_t - \boldsymbol{w}_s^* \cdot \boldsymbol{g}_s + \sigma_t \xi \right]^2 (\boldsymbol{v}_\mu \cdot \boldsymbol{g}_s)^2 \right\rangle . \end{aligned} \tag{71}$$

To simplify the first average, we note that it can be written

$$\langle \boldsymbol{u}_{n-1}^T [\boldsymbol{G}_s - \boldsymbol{g}_s \boldsymbol{g}_s^T] \boldsymbol{v}_\mu \boldsymbol{v}_\mu^T [\boldsymbol{G}_s - \boldsymbol{g}_s \boldsymbol{g}_s^T] \boldsymbol{u}_{n-1} \rangle = \langle (\boldsymbol{u}_{n-1} \cdot \boldsymbol{g}_s)^2 (\boldsymbol{v}_\mu \cdot \boldsymbol{g}_s)^2 \rangle - \left(\boldsymbol{u}_{n-1}^T \boldsymbol{G}_s \boldsymbol{v}_\mu \right)^2 . \tag{72}$$

Assuming $(\boldsymbol{u}_{n-1} \cdot \boldsymbol{g}_s)^2$ and $(\boldsymbol{v}_\mu \cdot \boldsymbol{g}_s)^2$ self average, for the first term we have

$$\left\langle (\boldsymbol{u}_{n-1} \cdot \boldsymbol{g}_s)^2 (\boldsymbol{v}_\mu \cdot \boldsymbol{g}_s)^2 \right\rangle = \boldsymbol{u}_{n-1}^T \boldsymbol{G}_s \boldsymbol{u}_{n-1} \, \boldsymbol{v}_\mu^T \boldsymbol{G}_s \boldsymbol{v}_\mu = \boldsymbol{u}_{n-1}^T \boldsymbol{G}_s \boldsymbol{u}_{n-1} \, \lambda_\mu . \tag{73}$$

Then, using the fact that $\boldsymbol{G}_s \boldsymbol{v}_\mu = \lambda_\mu \boldsymbol{v}_\mu$ and $\boldsymbol{u}_{n-1} \cdot \boldsymbol{v}_\mu = m_{\mu,n-1}$, we arrive at

$$\left\langle \boldsymbol{u}_{n-1}^T [\boldsymbol{G}_s - \boldsymbol{g}_s \boldsymbol{g}_s^T] \boldsymbol{v}_\mu \boldsymbol{v}_\mu^T [\boldsymbol{G}_s - \boldsymbol{g}_s \boldsymbol{g}_s^T] \boldsymbol{u}_{n-1} \right\rangle = \lambda_\mu \sum_\nu \lambda_\nu m_{\nu,n-1}^2 - \lambda_\mu^2 m_{\mu,n-1}^2 . \tag{74}$$

For the second average in Eq. (71), we again assume that $(\boldsymbol{v}_\mu \cdot \boldsymbol{g}_s)^2$ self averages, so the average of the product is just the product of the averages. The average of the square of the term in brackets is $\epsilon_{apr} + \sigma_t^2$ (see Eq. (10a)) and the average of $(\boldsymbol{v}_\mu \cdot \boldsymbol{g}_s)^2$ is, as in Eq. (73), λ_μ. Thus, the second average in Eq. (71) simplifies to

$$\left\langle \left[\boldsymbol{w}_t \cdot \boldsymbol{g}_t^n - \boldsymbol{w}_s^* \cdot \boldsymbol{g}_s^n + \sigma_t \xi \right]^2 (\boldsymbol{v}_\mu \cdot \boldsymbol{g}_s)^2 \right\rangle = (\epsilon_{apr} + \sigma_t^2) \lambda_\mu . \tag{75}$$

Inserting Eqs. (74) and (75) into (71), we arrive at

$$m_{\mu,n}^2 = (1 - 2\eta\lambda_\mu) m_{\mu,n-1}^2 + \eta^2 \lambda_\mu \sum_\nu \lambda_\nu m_{\nu,n-1}^2 + \eta^2 (\epsilon_{apr} + \sigma_t^2) \lambda_\mu . \tag{76}$$

To solve this equation, we define the matrix

$$A_{\mu\nu} \equiv 2\eta\lambda_\mu \delta_{\mu\nu} - \eta^2 \lambda_\mu \lambda_\nu . \tag{77}$$

After a small amount of algebra, we find that

$$\boldsymbol{m}_n^2 = \frac{\eta(\epsilon_{apr} + \sigma_t^2)}{2 - \eta D_0^s L_h} \mathbf{1}_h + (\boldsymbol{I} - \boldsymbol{A})^n \left(\boldsymbol{m}_0^2 - \frac{\eta(\epsilon_{apr} + \sigma_t^2)}{2 - \eta D_0^s L_h} \mathbf{1}_h \right) \tag{78}$$

where we used

$$\sum_\mu \lambda_\mu = D_0^s L_h \tag{79}$$

(which follows from Eq. (17)), $\boldsymbol{m}_n^2$ is a vector whose μ^{th} component is $m_{\mu,n}^2$, and we used the fact that $\boldsymbol{A}^{-1} \cdot \boldsymbol{\lambda} = \mathbf{1}_h / (2\eta - \eta^2 D_0^s L_h)$ where $\boldsymbol{\lambda} \equiv (\lambda_1, \lambda_2, ...)$, which follows from the Sherman-Morrison formula.

The term $(\boldsymbol{I} - \boldsymbol{A})^n$ is problematic, as its eigenvalues and eigenvectors cannot be found analytically. We thus make a very severe approximation: we let

$$A_{\mu\nu} \approx (2\eta - \eta^2 D_0^s L_h) \lambda_\mu \delta_{\mu\nu} . \tag{80}$$

With this approximation, Eq. (78) simplifies to

$$m_{\mu,n}^2 = \frac{\eta(\epsilon_{apr} + \sigma_t^2)}{2 - \eta D_0^s L_h} + \left(1 - \eta\lambda_\mu (2 - \eta D_0^s L_h)\right)^n \left(m_{\mu,0}^2 - \frac{\eta(\epsilon_{apr} + \sigma_t^2)}{2 - \eta D_0^s L_h} \right) . \tag{81}$$

We choose η to maximize the rate of decay of $m_{\mu,n}^2$ (that is, minimize $1 - \eta\lambda_\mu(2 - \eta D_0^s L_h)$); this yields

$$\eta^* = \frac{1}{D_0^s L_h} . \tag{82}$$

All the eigenmodes show the fastest decay at this learning rate regardless of their eigenvalues. Replacing η with η^* in Eq. (81), we have

$$m^2_{\mu,n} = m^2_\infty + \left(m^2_{\mu,0} - m^2_\infty\right)\left(1 - \frac{\lambda_\mu}{D^s_0 L_h}\right)^n \tag{83}$$

where

$$m^2_\infty \equiv \frac{\epsilon_{apr} + \sigma^2_t}{D^s_0 L_h}\,. \tag{84}$$

We can use this expression to determine how the estimation error, Eq. (69), evolves in time. Inserting Eq. (83) into (69), the average estimation error after n samples, $\bar{\epsilon}^{(n)}_{est}$, is given by

$$\bar{\epsilon}^{(n)}_{est} = \sum_\mu \left(\lambda_\mu m^2_\infty + \lambda_\mu (m^2_{\mu,0} - m^2_\infty)\left[1 - \frac{\lambda_\mu}{D^s_0 L_h}\right]^n\right)\,. \tag{85}$$

We now take advantage of the structure implicit in Eq. (34), which tells us that the eigenvalues are divided into four components, which we label with $q \in \{0, 1, 2, r\}$. We thus have

$$\bar{\epsilon}^{(n)}_{est} = \sum_{q\in\{0,1,2,r\}} \sum_{\mu\in S_q} \left(\lambda_\mu m^2_\infty + \lambda_\mu (m^2_{\mu,0} - m^2_\infty)\left[1 - \frac{\lambda_\mu}{D^s_0 L_h}\right]^n\right) \tag{86}$$

where S_q specifies the range of μ,

$$S_0\colon\ \mu = 1 \tag{87a}$$
$$S_1\colon\ 1 < \mu \le L_1 + 1 \tag{87b}$$
$$S_2\colon\ L_1 + 1 < \mu \le L_2 + L_1 + 1 \tag{87c}$$
$$S_r\colon\ L_2 + L_1 + 1 < \mu \le L_h\,. \tag{87d}$$

The first component, S_0, contains only one eigenmode, which corresponds to the largest eigenvalue λ_1 ($= \lambda^{(0)}$ in Eq. (35)). The rest contain multiple eigenmodes. For those modes we can approximate the exponential term as

$$(1 - \lambda_\mu/(D^s_0 L_h))^n \approx e^{-n\lambda_\mu/(D^s_0 L_h)} \approx e^{-n\langle\lambda_\mu\rangle_q/(D^s_0 L_h)} \tag{88}$$

where the subscript q means an average over $\mu \in S_q$. For the first inequality we used $\lambda_\mu \ll L_h$ for $\mu > 1$; for the second we used that fact that the eigenvalues typically have a small spread within each component. Making that replacement in Eq. (86), the lifetime cumulative estimation error, denoted $\bar{\epsilon}^N_{cml}$, is given by

$$\bar{\epsilon}^N_{cml} \equiv \frac{1}{N}\sum_{n=0}^{N-1} \bar{\epsilon}^n_{est} = \sum_{q\in\{0,1,2,r\}} \sum_{\mu\in S_q} \left(\lambda_\mu m^2_\infty + \lambda_\mu (m^2_{\mu,0} - m^2_\infty) R_q(L_h)\right) \tag{89}$$

where

$$R_q(L_h) \equiv \begin{cases} \frac{D^s_0 L_h}{N\lambda^{(0)}}\left[1 - \left(1 - \frac{\lambda^{(0)}}{D^s_0 L_h}\right)^N\right] & q = 0 \\ \frac{D^s_0 L_h}{N\langle\lambda_\mu\rangle_q}\left[1 - e^{-N\langle\lambda_\mu\rangle_q/(D^s_0 L_h)}\right] & \text{otherwise} \end{cases}\,. \tag{90}$$

The function $R_q(L_h)$ scales as $D^s_0 L_h/N\langle\lambda_\mu\rangle_q$ when $L_h \ll N\langle\lambda_\mu\rangle_q$ and approaches 1 when $L_h \gg N\langle\lambda_\mu\rangle_q$.

In §8 we computed the average eigenvalues (see Eq. (172)), so the only quantity we do not know is the average over $\lambda_\mu m^2_{\mu,0}$. That quantity is computed in the next section; using that result and applying a small amount of algebra, we arrive at

$$\bar{\epsilon}^N_{cml} = \sum_q L_q \langle\lambda_\mu\rangle_q \left[m^2_\infty\left(1 - R_q(L_h)\right) + \frac{\sigma^2_R}{L_h} R_q(L_h)\right] + \frac{(C^{ts}_q)^2}{C^{ss}_q}\left(1 - f_q(L_h)\right) R_q(L_h) \tag{91}$$

where

$$C^{ts}_r \equiv 0 \tag{92a}$$
$$L_0 \equiv 1\,, \tag{92b}$$

σ_R^2/L_h is the initial variance of the weights (Eq. (5)), and f_0, f_1 and f_2 are defined in Eq. (46). (Because $C_r^{ts}=0$, we do not need to define f_r.)

In Eq. (15) of the main text, we write down an expression for the average estimation error versus n. Here we derive that expression. As can be seen by comparing Eqs. (86) and (89) and taking into account the approximation made in Eq. (88), the only difference between $\bar{\epsilon}_{est}^{(n)}$ and $\bar{\epsilon}_{cml}^{N}$ is that $(1-\langle\lambda_\mu\rangle_q/D_0^sL_h)^n$ is replaced by $R_q(L_h)$. Making the reverse replacement in Eq. (89), and approximating $(1-\langle\lambda_\mu\rangle_q/D_0^sL_h)^n$ by $e^{-n\langle\lambda_\mu\rangle_q/D_0^sL_h}$, we have

$$\bar{\epsilon}_{est}^{n}=\epsilon_{apr}+\sigma_t^2+\sum_q\left[\frac{L_q\langle\lambda_\mu\rangle_q}{D_0^sL_h}\left(D_0^s\sigma_R^2-(\epsilon_{apr}+\sigma_t^2)\right)+\frac{(C_q^{ts})^2}{C_q^{ss}}\big(1-f_q(L_h)\big)\right]e^{-n\langle\lambda_\mu\rangle_q/(D_0^sL_h)}\,. \tag{93}$$

To derive this expression, we used the fact that $\sum_q L_q\langle\lambda_\mu\rangle_q=\sum_\mu\lambda_\mu=D_0^sL_h$ (see Eq. (79) for the second inequality), and we replaced m_∞^2 by $(\epsilon_{apr}+\sigma_t^2)/D_0^sL_h$ (see Eq. (84)). The terms in square brackets correspond to the b_q in Eq. (15) of the main text. The terms in the exponents were approximated from Eq. (37) and Eq. (172) as

$$\frac{\langle\lambda_\mu\rangle_q}{D_0^sL_h}\approx\frac{C_q^{ss}}{D_0^sL_q}\,, \tag{94}$$

where the expression for $q=2$ is valid in the regime $L_h\gg L_x$. For D_0^s and C_q^{ss} we used Eqs. (144) and (145), respectively.

Initial conditions

To estimate the contribution from the initial conditions (the term containing $\lambda_\mu m_{\mu,0}^2$ in Eq. (89)), we need an expression for $m_{\mu,0}$. Using $m_{\mu,0}=\boldsymbol{v}_\mu\cdot\boldsymbol{u}_0$, we write

$$m_{\mu,0}^2=\left(\boldsymbol{v}_\mu\cdot[\boldsymbol{w}_s^{(0)}-\boldsymbol{w}_s^*]\right)^2\approx\left(\boldsymbol{v}_\mu\cdot\boldsymbol{w}_s^{(0)}\right)^2+\left(\boldsymbol{v}_\mu\cdot\boldsymbol{w}_s^*\right)^2. \tag{95}$$

The projection weights are initialized as $\boldsymbol{w}_s^{(0)}\sim N(0,\sigma_R^2/L_h)$ (see Eq. (5)), so the first term is given approximately by

$$\left(\boldsymbol{v}_\mu\cdot\boldsymbol{w}_s^{(0)}\right)^2\approx\sum_{j=1}^{L_h}(v_{\mu,j})^2\left(\boldsymbol{w}_{s,j}^{(0)}\right)^2\approx\frac{\sigma_R^2}{L_h}\,. \tag{96}$$

For the second term we use Eq. (8) for $\boldsymbol{w}^*$, leading to

$$(\boldsymbol{v}_\mu\cdot\boldsymbol{w}_s^*)^2=\boldsymbol{v}_\mu^T\boldsymbol{G}_s^{-1}\langle\boldsymbol{g}_s\boldsymbol{g}_t^T\rangle\boldsymbol{w}_t\boldsymbol{w}_t^T\langle\boldsymbol{g}_t\boldsymbol{g}_s^T\rangle\boldsymbol{G}_s^{-1}\boldsymbol{v}_\mu\approx\frac{1}{L_t}\mathsf{Tr}\left[\boldsymbol{v}_\mu\boldsymbol{v}_\mu^T\boldsymbol{G}_s^{-1}\langle\boldsymbol{g}_s\boldsymbol{g}_t^T\rangle\langle\boldsymbol{g}_t\boldsymbol{g}_s^T\rangle\boldsymbol{G}_s^{-1}\right] \tag{97}$$

where the approximate expression follows from $\boldsymbol{w}_t\boldsymbol{w}_t^T\approx\boldsymbol{I}/L_t$. If we were to multiply the right hand side by λ_μ and sum over all μ, we would recover the left hand side of Eq. (43), because $\sum_\mu\lambda_\mu\boldsymbol{v}_\mu\boldsymbol{v}_\mu^T=\boldsymbol{G}_s$. Therefore, we can read off the sum of each component of μ from the right hand side of Eq. (43),

$$\sum_{\mu\in S_0}\lambda_\mu\left(\boldsymbol{v}_\mu\cdot\boldsymbol{w}_s^*\right)^2\approx\frac{(C_0^{ts})^2}{C_0^{ss}}\left[1-\frac{c_0}{c_0+L_h}\right] \tag{98a}$$

$$\sum_{\mu\in S_1}\lambda_\mu\left(\boldsymbol{v}_\mu\cdot\boldsymbol{w}_s^*\right)^2\approx\frac{(C_1^{ts})^2}{C_1^{ss}}\left[1-f\left(\frac{L_h}{L_x};c_1\right)\right] \tag{98b}$$

$$\sum_{\mu\in S_2}\lambda_\mu\left(\boldsymbol{v}_\mu\cdot\boldsymbol{w}_s^*\right)^2\approx\frac{(C_2^{ts})^2}{C_2^{ss}}\left[1-\frac{L_x}{L_h}\right]^+\left[1-f\left(\frac{L_h}{L_x^2/2};\frac{c_2}{1-L_x/L_h}\right)\right] \tag{98c}$$

$$\sum_{\mu\in S_r}\lambda_\mu\left(\boldsymbol{v}_\mu\cdot\boldsymbol{w}_s^*\right)^2\approx 0. \tag{98d}$$

5 Generalization error

To determine how the optimal hidden layer size, denoted L_h^*, scales with the input layer size, L_x, we need to minimize the generalization error (found by combining the approximation and estimation errors; see Eq. (9)) with respect to L_h. This is nontrivial: as can be seen in Figs. 4A and 5D, the optimum exhibits three different regimes, depending on the input

layer size, L_x. We can, though, access these regimes by considering different relative scaling of L_h and L_x: $L_h \gg L_x^2$, $L_x^2 \gg L_h \gg L_x$, and $L_x \gg L_h$. We begin by providing estimates for the approximation error, Eq. (45), in the three regimes; in the next two sections we use those results to compute the generalization error, first for maximum likelihood learning and then for stochastic gradient descent.

To see how the approximation error, Eq. (45), scales with L_h, note that (as mentioned after Eq. (44)) $f(\bar{\lambda}; c) \to c/\bar{\lambda}$ when $\bar{\lambda} \gg c$, and $f(\bar{\lambda}; c) \to 1$ when $\bar{\lambda} \to 0$. Using this, and the definitions of c_0, c_1 and c_2 in Eq. (36), it is straightforward to show that

$$\epsilon_{apr} + \sigma_t^2 \approx \begin{cases} \sigma_t^2 + \delta_{ts} + \left(\frac{C_2^{ts}}{C_2^{ss}}\right)^2 \frac{\delta_s L_x^2}{2L_h} & \text{if } L_h \gg L_x^2 \\ \sigma_t^2 + \delta_{ts} + \frac{(C_2^{ts})^2}{C_2^{ss}} + \left(\frac{C_1^{ts}}{C_1^{ss}}\right)^2 \frac{(\delta_s + C_2^{ss})L_x}{L_h} & \text{if } L_x^2 \gg L_h \gg L_x \\ \sigma_t^2 + \delta_{ts} + \frac{(C_2^{ts})^2}{C_2^{ss}} + \frac{(C_1^{ts})^2}{C_1^{ss}} + \left(\frac{C_0^{ts}}{C_0^{ss}}\right)^2 \frac{\delta_s + C_1^{ss} + C_2^{ss}}{c_0 + L_h} & \text{if } L_x \gg L_h \,. \end{cases} \tag{99}$$

We now use these expressions to compute the hidden layer size that optimizes the generalization error, first for maximum likelihood, and then for stochastic gradient descent.

5.1 Maximum likelihood

For maximum likelihood learning, the generalization error is given in Eq. (63). We need to combine that expression with Eq. (99), the approximation error, to get the generalization error, and minimize that with respect to L_h to find the optimal hidden layer size. Given the complexity of the generalization error, it is not possible to perform the exact minimization analytically. However, the generalization error becomes tractable in three regimes, $L_h \gg L_x^2$, $L_x^2 \gg L_h \gg L_x$ and $L_x \gg L_h$. We thus take the following four-step approach. In step 1, we assume that L_h lies in one of the regimes, say $L_h \ll L_x^2$ for definiteness. In step 2, we write down a simplified expression for the generalization error that is valid in that regime. In step 3, we find the value of L_h that minimizes the (simplified) generalization error. In step 4, we ask whether the minimum lies in the relevant region, in this case $L_h \ll L_x^2$. If it does, we have found a self-consistent minimum.

Optimal hidden layer size when $L_h \gg L_x^2$

In this regime, the generalization error is given by

$$\epsilon_{gen} \approx \left(\sigma_t^2 + \delta_{ts} + \left(\frac{C_2^{ts}}{C_2^{ss}}\right)^2 \frac{\delta_s L_x^2}{2L_h}\right) \frac{N}{N - L_h}. \tag{100}$$

Minimizing with respect to L_h yields

$$L_h^* = \sqrt{\left(B_2^{ml} L_x^2\right)^2 + B_2^{ml} N L_x^2} - B_2^{ml} L_x^2 \tag{101}$$

where

$$B_2^{ml} \equiv \left(\frac{C_2^{ts}}{C_2^{ss}}\right)^2 \frac{\delta_s}{2(\sigma_t^2 + \delta_{ts})}. \tag{102}$$

For this solution to be consistent with the condition $L_h \gg L_x^2$, N must satisfy $N \gg L_x^2/B_2^{ml}$. Therefore, if $L_x \ll \sqrt{B_2^{ml} N}$, then the hidden layer size that minimizes the generalization error is

$$L_h^* \approx \sqrt{B_2^{ml} N L_x^2}\,. \tag{103}$$

Optimal hidden layer size when $L_x^2 \gg L_h \gg L_x$

In this regime, the generalization error is given by

$$\epsilon_{gen} \approx \left(\sigma_t^2 + \delta_{ts} + \frac{(C_2^{ts})^2}{C_2^{ss}} + \left(\frac{C_1^{ts}}{C_1^{ss}}\right)^2 \frac{(\delta_s + C_2^{ss})L_x}{L_h}\right) \frac{N}{N - L_h}\,. \tag{104}$$

Minimizing with respect to L_h yields

$$L_h^* = \sqrt{(B_1^{ml} L_x)^2 + B_1^{ml} N L_x} - B_1^{ml} L_x \tag{105}$$

where

$$B_1^{ml} \equiv \left(\frac{C_1^{ts}}{C_1^{ss}}\right)^2 \frac{\delta_s + C_2^{ss}}{\sigma_t^2 + \delta_{ts} + (C_2^{ts})^2/C_2^{ss}} \,. \tag{106}$$

For this solution to be consistent with the condition $L_x^2 \gg L_h \gg L_x$, N must satisfy $L_x^3 \gg B_1^{ml} N \gg L_x$ Therefore, if $B_1^{ml} N \gg L_x \gg (B_1^{ml} N)^{1/3}$, then the hidden layer size that minimizes the generalization error is

$$L_h^* \approx \sqrt{B_1^{ml} N L_x} \,. \tag{107}$$

Optimal hidden layer size when $L_x \gg L_h$

In this regime, the generalization error is given by

$$\epsilon_{gen} \approx \left(\sigma_t^2 + \delta_{ts} + \frac{(C_1^{ts})^2}{C_1^{ss}} + \frac{(C_2^{ts})^2}{C_2^{ss}} + \left(\frac{C_0^{ts}}{C_0^{ss}}\right)^2 \frac{\delta_s + C_1^{ss} + C_2^{ss}}{c_0 + L_h}\right) \frac{N}{N - L_h} \,. \tag{108}$$

Minimizing with respect to L_h yields

$$L_h^* = \sqrt{(B_0^{ml})^2 + B_0^{ml}(N + c_0)} - B_0^{ml} - c_0 \tag{109}$$

where

$$B_0^{ml} \equiv \left(\frac{C_0^{ts}}{C_0^{ss}}\right)^2 \frac{\delta_s + C_1^{ss} + C_2^{ss}}{\sigma_t^2 + \delta_{ts} + (C_1^{ts})^2/C_1^{ss} + (C_2^{ts})^2/C_2^{ss}} \,. \tag{110}$$

For this solution to be consistent with the condition $L_x \gg L_h$, N must satisfy $L_x \gg \sqrt{B_0^{ml} N}$. Assuming also that $N \gg 1$, we have

$$L_h^* \approx \sqrt{B_0^{ml} N}, \tag{111}$$

Here the optimal hidden layer size, L_h^*, does not depend on the input layer size, L_x.

Optimal hidden layer size when $N \propto L_x^\gamma$

Based on empirical observations (Table 2 and Fig. S4), we found that $N \propto L_x^\gamma$, with γ between about 1.6 and 2. Here, we combine this result with the three scaling derived above to determine how the optimal hidden layer size depends on γ. We find the following:

1. $L_h \gg L_x^2$: We see from Eq. (103) that $L_h^* \propto L_x^{1+\gamma/2}$. To ensure self-consistency, we must have $N \gg L_x^2$ (see comments preceding Eq. (103)), which, combined with $N \propto L_x^\gamma$, requires $\gamma > 2$.

2. $L_x^2 \gg L_h \gg L_x$: We see from Eq. (107) that $L_h^* \propto L_x^{1/2+\gamma/2}$. To ensure self-consistency, we must have $L_x^3 \gg N \gg L_x$ (see comments preceding Eq. (107)), which, combined with $N \propto L_x^\gamma$, requires $1 < \gamma < 3$.

3. $L_x \gg L_h$: We see from Eq. (111) that $L_h^* \propto L_x^{\gamma/2}$. To ensure self-consistency, we must have $L_x^2 \gg N$ (see comments preceding Eq. (111)), which, combined with $N \propto L_x^\gamma$, requires $\gamma < 2$.

When $\gamma > 3$, the network must operate in regime (1), while when $\gamma < 1$, the network must operate in regime (3). When γ is between 1 and 3, on the other hand, the network can operate in two regimes: when $1 < \gamma < 2$, either (2) or (3); and when $2 < \gamma < 3$, either (1) and (2). However, the one with steeper scaling between L_x and L_h^* has smaller error, and so generates the relevant scaling. To see why, note that in regimes (1), (2) and (3), the generalization error (Eqs. (100), (104) and (108), respectively) is given approximately by

$$\text{regime (1):}\ \epsilon_{gen} \approx \sigma_t^2 + \delta_{ts} \tag{112a}$$

$$\text{regime (2):}\ \epsilon_{gen} \approx \sigma_t^2 + \delta_{ts} + \frac{(C_2^{ts})^2}{C_2^{ss}} \tag{112b}$$

$$\text{regime (3):}\ \epsilon_{gen} \approx \sigma_t^2 + \delta_{ts} + \frac{(C_1^{ts})^2}{C_1^{ss}} + \frac{(C_2^{ts})^2}{C_2^{ss}} \,. \tag{112c}$$

Consequently, regime (1) is favored over regime (2), and regime (2) is favored over regime (3).

In summary, our analytical results indicates that as a function of γ, the exponent of the scaling law should follow the black line in Fig. 4F. Minimizing Eq. (63) numerically, we indeed found that this is the case (blue line in Fig. S7A). In this context, the 3/2-law is somewhat special, in the sense that a scaling factor between 3/2 to 2 is not feasible in our model setting. In addition, this result, combined with the observation that γ is below 2, indicates that 7/2 scaling seen among insects is also not feasible unless there is an additional constraint.

5.2 Stochastic Gradient Descent

For stochastic gradient descent, we use the cumulative generalization error, denoted ϵ_{cg}^N and defined to be

$$\epsilon_{cg}^N \equiv \frac{1}{N}\sum_{n=0}^{N-1}\left(\epsilon_{apr} + \sigma_t^2 + \bar{\epsilon}_{est}^{(n)}\right) = \epsilon_{apr} + \sigma_t^2 + \bar{\epsilon}_{cml}^N \,. \tag{113}$$

Using Eq. (91) for $\bar{\epsilon}_{cml}^N$, and combining that with Eq. (84) for m_∞^2 and then applying Eq. (79) to simplify the resulting expression, we arrive, after a small amount of algebra, at

$$\epsilon_{cg}^N = (\epsilon_{apr} + \sigma_t^2)\left[2 - \frac{1}{D_0^s L_h}\sum_q L_q \langle\lambda_\mu\rangle_q R_q(L_h)\right] + \sum_q R_q(L_h)\left[L_q\langle\lambda_\mu\rangle_q \frac{\sigma_R^2}{L_h} + \frac{(C_q^{ts})^2}{C_q^{ss}}\big(1 - f_q(L_h)\big)\right] . \tag{114}$$

The critical quantity in this equation is $\langle\lambda_\mu\rangle_q$, which is given in Eq. (172) (but with slightly different notation). We repeat that equation here, following the notation used in §4.2, with a focus on the behavior when L_h is either very small or very large,

$$\langle\lambda_\mu\rangle_0 \approx C_0^{ss} L_h \qquad L_h \gg 1 \tag{115a}$$

$$\langle\lambda_\mu\rangle_1 \approx \begin{cases} C_1^{ss} L_h / L_x & L_h \gg L_x \\ \delta_s + C_1^{ss} + C_2^{ss} & L_h \ll L_x \end{cases} \tag{115b}$$

$$\langle\lambda_\mu\rangle_2 \approx \begin{cases} 2C_2^{ss} L_h / L_x^2 & L_h \gg L_x^2 \\ \delta_s + C_2^{ss}\left[1 - L_x/L_h\right]^+ & L_h \ll L_x^2 \end{cases} \tag{115c}$$

$$\langle\lambda_\mu\rangle_r = \delta_s \,. \tag{115d}$$

To make it easier to analyze the generalization error, it is convenient to use Eq. (90) to express $R_q(L_h)$ in terms of more fundamental quantities, yielding

$$\begin{aligned}\epsilon_{cg}^N \approx (\epsilon_{apr} + \sigma_t^2)\left[2 - \sum_{q\neq 0}\frac{L_q}{N}\left[1 - e^{-N\langle\lambda_\mu\rangle_q / D_0^s L_h}\right]\right] + \frac{D_0^s}{N}\left[\sigma_R^2 + \frac{(C_0^{ts})^2}{(C_0^{ss})^2} - \frac{\epsilon_{apr} + \sigma_t^2}{D_0^s}\right] \\ + \sum_{q\neq 0}\frac{D_0^s L_q}{N}\left[1 - e^{-N\langle\lambda_\mu\rangle_q / D_0^s L_h}\right]\left[\sigma_R^2 + \frac{(C_q^{ts})^2}{C_q^{ss}}\frac{L_h}{L_q\langle\lambda_\mu\rangle_q}\big(1 - f_q(L_h)\big)\right] .\end{aligned} \tag{116}$$

To derive this expression, we replaced $\langle\lambda_\mu\rangle_0$ with $C_0^{ss} L_h$ (Eq. (115a)) and $f_0(L_h)$ with c_0/L_h (Eq. (46a) in the large L_h limit), used the fact that $L_0 = 1$ (Eq. (92b)), assumed $N \gg 1$, and replaced $(1 - (C_0^{ss}/D_0^s))^N$ with 0, which is valid in the large N limit.

In the following subsections, we minimize ϵ_{cg}^N with respect to L_h to find the optimal hidden layer size. As with maximum likelihood, we work in three different regimes, and again in each of them the estimation error becomes tractable. To simplify our analysis, we make assumptions about N that assures the solution in each region is self-consistent.

Optimal hidden layer size when $L_h \gg L_x^2$

We assume that $N \gg L_x^2$, which will yield a self-consistent solution, as we show below. In this regime, Eq. (37) tells us that $L_1 = L_x, L_2 \approx L_x^2/2$, and $L_r \approx L_h$. Consequently, using Eq. (90), with average eigenvalues given by Eq. (115), we see that $R_0(L_h)$, $R_1(L_h)$ and $R_2(L_h)$ are all approximately zero, and

$$R_r(L_h) = \frac{D_0^s L_h}{\delta_s N}\left(1 - e^{-\frac{\delta_s N}{D_0^s L_h}}\right). \tag{117}$$

Inserting this into Eq. (114), using Eq. (115d) for $\langle\lambda_\mu^{(r)}\rangle$, recalling that $C_r^{ts} = 0$ (see Eq. (92a)), and using the fact that all the other R_q are approximately zero, we see that the cumulative generalization error is given approximately by

$$\epsilon_{cg}^N \approx \big(\epsilon_{apr} + \sigma_t^2\big)\left(2 - \frac{L_h}{N}\left[1 - e^{-\frac{\delta_s N}{D_0^s L_h}}\right]\right) + \frac{\sigma_R^2 D_0^s L_h}{N}\left(1 - e^{-\frac{\delta_s N}{D_0^s L_h}}\right). \tag{118}$$

The first term is a monotonically decreasing function of L_h while the second term is monotonically increasing. When σ_R^2 is too small, the second term becomes too weak to supports the presence of the non-trivial minimum (gray points

in Fig. 5F). However, this initial weight dependence can be avoided by using an adaptive learning rate (black points in Fig. 5F), although the analytical estimation of the error becomes difficult in that case.

When $N \ll L_h$, the L_h dependence in all but the term ϵ_{apr} in Eq. (118) disappears. We thus consider the opposite limit, $N \gg L_h$. Then, using Eq. (99) for the approximation error, we find, in this limit, that

$$\epsilon_{cg}^N \approx 2(\delta_{ts} + \sigma_t^2) + \left(\frac{C_2^{ts}}{C_2^{ss}}\right)^2 \frac{\delta_s L_x^2}{L_h} + \left(D_o^s \sigma_R^2 - [\delta_{ts} + \sigma_t^2]\right) \frac{L_h}{N} \,. \tag{119}$$

Minimizing with respect to L_h gives

$$L_h^* = B_2^{sgd} \sqrt{N L_x^2}, \tag{120}$$

where

$$B_2^{sgd} \equiv \frac{C_2^{ts}}{C_2^{ss}} \sqrt{\frac{\delta_s}{D_0^s \sigma_R^2 - (\delta_{ts} + \sigma_t^2)}}. \tag{121}$$

We need to check for self-consistency, which means we need to check that $N \gg L_h^*$ and $L_h^* \gg L_x^2$. For the first, we combine the condition $L_x^2 \ll N$ with Eq. (120) to obtain $L_h^* \ll N$ (note that B_2^{sgd} is $\mathcal{O}(1)$). For the second, we combine the condition $N \gg L_x^2$ with Eq. (120) to obtain $L_h^* \gg L_x^2$. Thus, this is a self-consistent solution.

Optimal hidden layer size when $L_x^2 \gg L_h \gg L_x$

In this regime we assume that $N \gg L_x$, which, as we show below, will again yield a self-consistent solution. Notably, this assumption is consistent with the assumed scaling in Fig. 5, $N \propto L_x^{1.9}$. In the regime $L_x^2 \gg L_h \gg L_x$, Eq. (37) tells us that $L_1 = L_x, L_2 \approx L_h$, and $L_r = 0$. Consequently, using Eq. (90), with average eigenvalues given by Eq. (115), we see that $R_0(L_h)$ and $R_1(L_h)$ are approximately zero, and

$$R_2(L_h) = \frac{D_0^s L_h}{N \langle \lambda_\mu^{(2)} \rangle} \left[1 - e^{-N \langle \lambda_\mu^{(2)} \rangle / (D_0^s L_h)}\right] . \tag{122}$$

Inserting this into Eq. (114), using Eq. (115c) for $\langle \lambda_\mu^{(2)} \rangle$, noting that $f_2(L_h) \approx f(2L_h/L_x^2; c_2) \approx 1$ (see Eq. (46c)), and using the fact that all the other R_q are approximately zero, the cumulative generalization error is given approximately by

$$\epsilon_{cg}^N \approx (\epsilon_{apr} + \sigma_t^2) \left(2 - \frac{L_h}{N} \left[1 - e^{-\frac{(\delta_s + C_2^{ss})N}{D_o^s L_h}}\right]\right) + \frac{\sigma_R^2 D_o^s L_h}{N} \left[1 - e^{-\frac{(\delta_s + C_2^{ss})N}{D_0^s L_h}}\right] . \tag{123}$$

Following the arguments in the previous section, we consider the limit $N \gg L_h$. Then, using Eq. (99) for the approximation error, we find, in this limit, that

$$\epsilon_{cg}^N \approx 2 \left(\delta_{ts} + \sigma_t^2 + \frac{(C_2^{ts})^2}{C_2^{ss}}\right) + 2 \left(\frac{C_1^{ts}}{C_1^{ss}}\right)^2 \frac{(\delta_s + C_2^{ss}) L_x}{L_h} + \left(D_o^s \sigma_R^2 - \left[\delta_{ts} + \sigma_t^2 + \frac{(C_2^{ts})^2}{C_2^{ss}}\right]\right) \frac{L_h}{N} \,. \tag{124}$$

Minimizing with respect to L_h yields

$$L_h^* = B_1^{sgd} \sqrt{N L_x} \,, \tag{125}$$

where

$$B_1^{sgd} \equiv \frac{C_1^{ts}}{C_1^{ss}} \sqrt{\frac{2(\delta_s + C_2^{ss})}{D_o^s \sigma_R^2 - [\delta_{ts} + (C_2^{ts})^2 / C_2^{ss} + \sigma_t^2]}} \,. \tag{126}$$

We need to check for self-consistency, which means we need to check that $N \gg L_h^*$ and $L_x^2 \gg L_h^* \gg L_x$. For the first, we combine the condition $L_x \ll N$ with Eq. (125) to obtain $L_h^* \ll N$ (note that B_1^{sgd} is $\mathcal{O}(1)$). For the second, we combine the condition $N \gg L_x$ with Eq. (125) to obtain $L_h^* \gg L_x$. Thus, this is a self-consistent solution. To ensure $L_x^2 \gg L_h^*$, N needs to satisfy $N \ll L_x^3$. Thus, N must be large but not too large.

Optimal hidden layer size when $L_x \gg L_h$

In this regime, we assume that $N \gg 1$, which is again consistent with the scaling in Fig. 5, $N \propto L_x^{1.9}$. In the regime $L_x \gg L_h$, Eq. (37) tells us that $L_1 = L_h$ and $L_2 = L_r = 0$. Because L_2 and L_r are zero and $N \gg 1$, $q = 1$ is the only relevant term, so cumulative generalization error, Eq. (114), is given approximately by

$$\epsilon_{cg}^N \approx (\epsilon_{apr} + \sigma_t^2)\left(2 - \frac{L_h}{N}\left[1 - e^{-\frac{\langle\lambda^{(1)}\rangle N}{D_0^s L_h}}\right]\right) + \frac{\sigma_R^2 D_0^s L_h}{N}\left(1 - e^{-\frac{\langle\lambda^{(1)}\rangle N}{D_0^s L_h}}\right). \tag{127}$$

As before, using Eq. (99) and assuming $N \gg L_h$, the above equation becomes

$$\epsilon_{cg}^N \approx 2\left(\frac{C_0^{ts}}{C_0^{ss}}\right)^2 \frac{\delta_s + C_1^{ss} + C_2^{ss}}{L_h} + \left(D_0^s\sigma_R^2 - \left[\sigma_t^2 + \delta_{ts} + \frac{(C_1^{ts})^2}{C_1^{ss}} + \frac{(C_2^{ts})^2}{C_2^{ss}}\right]\right)\frac{L_h}{N} + \text{const.} \tag{128}$$

Thus, the optimal hidden layer size is given by

$$L_h^* = B_0^{sgd}\sqrt{N}, \tag{129}$$

where

$$B_0^{sgd} \equiv \frac{C_0^{ts}}{C_0^{ss}}\sqrt{\frac{2(\delta_s + C_1^{ss} + C_2^{ss})}{D_0^s\sigma_R^2 - [\delta_{ts} + (C_1^{ts})^2/C_1^{ss} + (C_2^{ts})^2/C_2^{ss} + \sigma_t^2]}}. \tag{130}$$

This is a self-consistent solution at $L_x^2 \gg N$.

Optimal hidden layer size when $N \propto L_x^\gamma$

As we did under MLE above, we combine these three scalings with the emperical observation that $N \propto L_x^\gamma$ to determine how the optimal hidden layer size depends on γ. We find the following:

1. $L_h \gg L_x^2$ and $N \gg L_x^2$: We see from Eq. (120) that $L_h^* \propto L_x^{1+\gamma/2}$. In addition, combining $N \gg L_x^2$ with $N \propto L_x^\gamma$, we see that $\gamma > 2$.

2. $L_x^2 \gg L_h \gg L_x$, $L_x^3 \gg N$ and $N \gg L_x$: We see from Eq. (125) that $L_h^* \propto L_x^{1/2+\gamma/2}$. In addition, combining $N \gg L_x$ and $L_x^3 \gg N$ with $N \propto L_x^\gamma$, we see that $3 > \gamma > 1$.

3. $L_x \gg L_h$ and $L_x^2 \gg N$: We see from Eq. (129) that $L_h^* \propto L_x^{\gamma/2}$. In addition, combining $L_x^2 \gg N$ with $N \propto L_x^\gamma$, we see that $\gamma < 2$.

As with MLE, when $\gamma > 3$ or $\gamma < 1$ the network can operate in only one regime, but when $1 < \gamma < 3$ it can operate in two. Also as with MLE, the one with steeper scaling between L_x and L_h^* has smaller error, and so generates the relevant scaling. To see why, note that to leading order, the generalization error is given by $2(\epsilon_{apr} + \sigma_t^2)$ (see Eqs. (118), (123) and (127)), and from Eq. (99) we see that the generalization error increases from regime (1) to regime (2) to regime (3). Consequently, regime (1) is favored over regime (2), and regime (2) is favored over regime (3).

In summary, our analytical results indicates that as a function of γ, the exponent of the scaling law should follow the black line in Fig. 4F. Minimizing Eq. (113) numerically, we indeed found that this was the case (blue line in Fig. S7B).

Optimal hidden layer size when the learning is terminated after αN samples

In Fig. S8, we considered the case when the learning by SGD is terminated after αN samples. In this case, the cumulative error over all N samples is given by,

$$\epsilon_\alpha = \alpha\epsilon_{cg}^{(\alpha N)} + (1-\alpha)\epsilon_{gen}^{(\alpha N)}, \tag{131}$$

where $\epsilon_{cg}^{(\alpha N)}$ is the cumulative error over $n = 1, ..., \alpha N$ (Eq. 113), and $\epsilon_{gen}^{(\alpha N)} = \epsilon_{est}^{(\alpha N)} + \epsilon_{apr} + \sigma_t^2$, is the generalization error at $n = \alpha N$, where $\epsilon_{est}^{(\alpha N)}$ and ϵ_{apr} are defined at Eqs. 85 and 45.

6 Model with low precision hard-wired connections

In our analysis, we assume that the initial weights are random. However, weights can also be tuned on evolutionary timescales. To model this, we add a parallel hidden layer corresponding to lateral horn neurons (see Fig. 6A),

$$y = \boldsymbol{w}_p \cdot g(\boldsymbol{J}_p \boldsymbol{x}) + \boldsymbol{w}_s \cdot g(\boldsymbol{J}_s \boldsymbol{x}) \tag{132}$$

where g is ReLU, $\boldsymbol{J}_p \in \Re^{L_p \times L_x}$, and $\boldsymbol{w}_p \in \Re^{L_p}$. Motivated by the suggestion that the connections from the projection neurons to lateral horn neurons are genetically specified [27], we assume that $\boldsymbol{J}_p$ and $\boldsymbol{w}_p$ are genetically encoded, and those weights are tuned over evolutionary timescales. If the weights were tuned perfectly, they would be set to

$$\boldsymbol{J}_p^*, \boldsymbol{w}_p^* = \arg\min_{\boldsymbol{J}_p, \boldsymbol{w}_p} \left\langle [\boldsymbol{w}_t \cdot g(\boldsymbol{J}_t \boldsymbol{x}) - \boldsymbol{w}_p \cdot g(\boldsymbol{J}_p \boldsymbol{x})]^2 \right\rangle_{p(\boldsymbol{x})} . \tag{133}$$

However, there are two problems with setting the weights to $\boldsymbol{J}_p^*$ and $\boldsymbol{w}_p^*$. One is that this would requires an infinite number of bits, while the genetic capacity is limited. The other is that evolution cannot know perfectly the odors an animal will encounter, or their valances, making it impossible to compute exactly the average on the right hand side of Eq. (133). Thus, the weights of the hard-wired the lateral horn pathway cannot be specified perfectly.

To take this into account, we set the weights $\boldsymbol{J}_p$ and $\boldsymbol{w}_p$ in two steps. In the first step, we find the optimal weights via Eq. (133). In the second we corrupt the weights, either by adding noise to them or by discretizing them. Given a network, the amount of corruption is determined by what we call the genetic budget, denoted G. Below, we describe how we implement these two steps.

6.1 Implementation of the genetic budget

We will assume that the genome effectively supplies s_b bits per synapse. In Sec. 6.3 we discuss how each synapse is corrupted given s_b; here we simply compute the genetic budget, G, in terms of this quantity. For that we count the number of bits it takes to specify the weights. Assuming the weights are independent and random, the total information required to wire up the circuit is $s_b\times$ [the number of synapses]. There are $L_p \times L_x$ synapses in the matrix $\boldsymbol{J}_p$ and L_p synapses in the vector $\boldsymbol{w}_p$, giving us a total of $(L_p L_x + L_p)$ synapses. Thus, to wire up this circuit innately requires

$$G = L_p(L_x + 1)s_b \tag{134}$$

bits of information to be stored in the genome. This gives us the constraint, Eq. (18), we used in the main text.

This estimate assumes that the weights do not contain structure that can be exploited by the genome. If they do – that is, if the true weights are compressible – then the amount of genetic material required for encoding can be smaller than G in Eq. (134). For instance, if the weights $\boldsymbol{J}_p$ are sparsely distributed (something we consider in the next section), Eq. (134) would not apply. More generally, the minimum genome size can be computed by estimating the Kolmogorov complexity, which is the minimum length of a program that generates $\boldsymbol{J}_p$ and $\boldsymbol{w}_p$. This is indeed a more relevant measure of genetic capacity, as the genome does not transmit information like a communication channel; instead, it is more like a program that is run to construct the connectome. However, here we simply make the assumption that the weights contain very little structure, and in particular are approximately random and uncorrelated. This assumption is reasonable in our model setting, because the weights of the teacher network is randomly generated, and the circuit size of the genetically specified pathway is much smaller than the size of the teacher network. However, this assumption might be violated in the actual brain. In this regime, the average Kolmogorov complexity is the same as the entropy, up to constant [28]. Thus, Eq. (134) should provide a reasonable bound on the Kolmogorov complexity. In simulations, we fixed G/s_b to a constant, then changed s_b and L_x. This means that the genetic-pathway size $L_p = G/\left(s_b(L_x+1)\right)$ decreases monotonically as a function of L_x.

6.2 A model with sparse connectivity in the genetically-specified pathway

As mentioned above, it might be possible to achieve a more efficient encoding by using a sparsely connected network, as that requires less information to specify the weights (simply because there are fewer of them). However, whether or not a sparse network is more efficient depends on the degree to which sparseness affects expressivity.

To test this idea, we constructed a genetically specified pathway for which the connectivity matrix, $\boldsymbol{J}_p$ was sparse, with the sparsity given by the parameter ρ_s,

$$\rho_s \equiv \mathrm{prob}[J_{ij}^p \neq 0] . \tag{135}$$

Assuming, as above, that the genome can effectively supply s_b bits per synapses, the total information it needs to supply is $L_p(\rho_s L_x + 1)s_b$ bits. Then, because it takes $H(\rho_s)L_pL_x$ bits to specify which weights are nonzero, where $H(\rho_s)$ is the entropy of a Bernoulli random variable with probability ρ_s, the total genetic budget, G, is given by

$$G = L_p \left(L_x[\rho_s s_b + H(\rho_s)] + s_b \right) . \tag{136}$$

In the limit $\rho_s \to 1$, we recover Eq. (134). Under a fixed genetic budget, G, and a fixed number of bits per synapses, s_b, the genetically specified hidden layer size, L_p, becomes a function of L_x,

$$L_p = \frac{G}{L_x[\rho_s s_b + H(\rho_s)] + s_b} . \tag{137}$$

In our simulations we fix s_b, G and ρ_s, and let L_p depend on L_x via Eq. (137).

In Fig. S9B, we numerically estimated the optimal hidden layer size of the developmentally learned pathway, L_h, for a range of sparsity, ρ_s. The optimization was performed as described in §7.5, except that after we optimized $\boldsymbol{J}_p^*$ for $m = 2.5 \times 10^5$ steps, we sparsified $\boldsymbol{J}_p^*$ by keeping the $\rho_s L_x$ largest weights (in terms of the absolute weight) for each postsynaptic neuron, and setting the rest to zero. We then retrained the non-zero elements of J_p and all the elements of w_p for another $m = 2.5 \times 10^5$ steps. Finally, we created s_b bit representations by discretizing the weights. Here, we discretized the positive and the negative weights of $\boldsymbol{J}_p^*$ separately, because most elements of $\boldsymbol{J}_p^*$ are zero.

Under moderate sparseness ($\rho_s = 0.5$), the genetic-pathway approximates the teacher model better than a fully connected one with the same genetic budget G. As a result, the optimal hidden layer size of the developmental pathway, L_h^*, was slightly smaller that of the vanilla model (dark green vs black lines in Fig. S9B). In a sparser circuit, the optimal hidden layer size L_h is similar or slightly larger than in the fully connected model (light green line; $\rho_c = 0.25$), indicating that the genetic pathway is less effective, potentially due to sub-optimal convergence.

6.3 Setting the weights of the hard-wired connections

As discussed above, we set the weights of hard-wired connections, $\boldsymbol{J}_p$ and $\boldsymbol{w}_p$, by finding their optimal values, $\boldsymbol{J}_p^*$ and $\boldsymbol{w}_p^*$, then compressing each weight to at most s_b bits. To ensure that our results were robust, we used two different methods for compressing the weights. They are described below .

6.3.1 Numerical estimation of $\boldsymbol{J}_p^*$ and $\boldsymbol{w}_p^*$

We used a mini-batch backpropagation for updating $\boldsymbol{J}_p$,

$$\boldsymbol{J}_p^{(m+1)} = \boldsymbol{J}_p^{(m)} - \eta \sum_{b=1}^{B} \frac{\partial}{\partial \boldsymbol{J}_p} \left(\boldsymbol{w}_p \cdot g(\boldsymbol{J}_p \boldsymbol{x}_b) - \boldsymbol{w}_t \cdot g(\boldsymbol{J}_t \boldsymbol{x}_b) \right)^2 , \tag{138}$$

where m is the update count. We used mini-batch size $B = 2000$, learning rate $\eta = 0.005$, and the teacher noise was excluded from the supervised signal to achieve fast convergence. Because the optimization of $\boldsymbol{J}_p$ is an evolutionary process, the update rule does not need to be local. The weight, $\boldsymbol{w}_p$, was updated after each minibatch update of $\boldsymbol{J}_p$ according to

$$\boldsymbol{w}_p^{(m)} \equiv \left\langle g(\boldsymbol{J}_p^{(m)} \boldsymbol{x}) g(\boldsymbol{J}_p^{(m)} \boldsymbol{x})^T \right\rangle^{-1} \left\langle g(\boldsymbol{J}_p^{(m)} \boldsymbol{x}) g(\boldsymbol{J}_t \boldsymbol{x})^T \right\rangle \boldsymbol{w}_t . \tag{139}$$

The above expectations, which are over $\boldsymbol{x}$, were obtained analytically using Eq. (153) below. We initialized $\boldsymbol{J}_p^{(m=0)}$ to $J_{ij}^{p,0} \sim N(0, 1/L_x)$, then updated $\boldsymbol{w}_p$ and $\boldsymbol{J}_p$ alternatively for 10^5 steps (see §7.5 for an algorithmic description), which was typically enough to achieve convergence.

6.3.2 Compression of the weights by discretization

Synaptic weights can be compressed by discretization or by adding noise [29]. Although, we mainly used the latter, we describe the discretization-based approach first, as it is more intuitive.

Real-valued weights, w_j^*, can be compressed to s_b bits by simply discretizing them into 2^{s_b} equal probability states. We used instead states with unequal probabilities, so we compress to slightly less than s_b bits. Denoting $w_{max} \equiv \max\{w_1^*, ..., w_{L_p}^*\}$, $w_{min} \equiv \min\{w_1^*, ..., w_{L_p}^*\}$, and $\Delta w \equiv (w_{max} - w_{min})/(2^{s_b})$, compressed weights, w_j, are obtained via

$$w_j = w_{min} + \left(\left\lfloor \frac{w_j^* - w_{min}}{\Delta w} \right\rfloor + \frac{1}{2} \right) \Delta w, \tag{140}$$

where $\lfloor x \rfloor$ returns the largest integer less than or equal to x. Both the hidden and output weights, $\boldsymbol{J}_p$ and $\boldsymbol{w}_p$, can be compressed in this manner.

6.3.3 Compression of the weights by adding noise

Assuming that the optimal weight w_j^* is sampled from a Gaussian distribution (i.e., $w_j^* \sim N(0, \sigma_w^2)$), the bit length required for encoding the weight can be reduced by shrinking the weight while adding noise,

$$w_j = \sqrt{1-\gamma^2} w_j^* + \gamma \sigma_w \zeta, \tag{141}$$

where ζ is a zero mean, unit variance Gaussian variable, and γ is the relative noise amplitude ($0 \le \gamma \le 1$). Marginalizing over w_j^*, we get $w_j \sim N(0, \sigma_w^2)$. Thus, the mutual information between w_j and w_j^* is

$$I[w_j; w_j^*] = -\log \gamma. \tag{142}$$

Therefore, to compress the weight into s_b bits, γ needs to be set to $\gamma = e^{-(\log 2) s_b}$ ($\log 2$ is for the conversion from nats to bits). Note that in this formulation, 2^{s_b} does not need to be an integer, unlike for the discretization method.

In the simulations, we estimated the variance of $\{w_j^*\}$ and $\{J_{ij}^*\}$ numerically, after they were optimized. Denoting the variances as σ_w^2 and σ_J^2, the compressed weights are given as

$$w_j = \sqrt{1-\gamma^2} w_j^* + \gamma \sigma_w \zeta \quad \text{and} \quad J_{ij} = \sqrt{1-\gamma^2} J_{ij}^* + \gamma \sigma_J \zeta. \tag{143}$$

7 Details of the numerical analysis

7.1 ReLU activation

We first compute the parameters when both the teacher and student use ReLU activation ($g_t(u) = g_s(u) = \max(0, u)$). The diagonal elements, Eqs. (17) and (39), are given by

$$D_0^s = D_0^t = \int_0^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du = \frac{1}{2}. \tag{144}$$

The off diagonal elements, Eqs. (21) and (33), are given by

$$C_0^{ts} = C_0^{ss} = \frac{1}{2\pi} \tag{145a}$$

$$C_1^{ts} = C_1^{ss} = \frac{1}{4} \tag{145b}$$

$$C_2^{ts} = C_2^{ss} = \frac{1}{4\pi}. \tag{145c}$$

In this setting, the coefficient for the third order term is identically zero, so the second-order approximation effectively achieves third-order accuracy. Inserting the coefficients into Eq. (45), the approximation error is estimated as

$$\begin{aligned} \epsilon_{apr} \approx\ & \delta_s + \frac{1}{4\pi}\left(1 - \left[1 - \frac{L_x}{L_h}\right]^+\right) + \frac{\pi - 1}{2\pi(\pi - 1 + L_h)} \\ & + \frac{1}{8}\left[\sqrt{\left(\frac{L_h}{L_x} + 4\delta_s + \frac{1}{\pi} - 1\right)^2 + 4\left(4\delta_s + \frac{1}{\pi}\right)} - \left(\frac{L_h}{L_x} + 4\delta_s + \frac{1}{\pi} - 1\right)\right] \\ & + \frac{1}{8\pi}\left[1 - \frac{L_x}{L_h}\right]^+ \left[\sqrt{\left(\frac{2L_h}{L_x^2} + \frac{4\pi\delta_s}{1 - L_x/L_h} - 1\right)^2 + \frac{16\pi\delta_s}{1 - L_x/L_h}} - \left(\frac{2L_h}{L_x^2} + \frac{4\pi\delta_s}{1 - L_x/L_h} - 1\right)\right] \end{aligned} \tag{146}$$

where $\delta_s = (\pi - 3)/4\pi$ (see Eq. (27)).

7.2 Logistic activation

We next consider the case of model mismatch, where the teacher activation function is ReLU but the student is a logistic function, $g_s(u) = 1/(1 + e^{-u})$. The diagonal element of the teacher, D_0^t, is the same as above, but the student is different,

$$D_0^s = \int_{-\infty}^\infty \frac{du}{\sqrt{2\pi}} \left(\frac{1}{1 + e^{-u}}\right)^2 \exp\left(-\frac{u^2}{2}\right) \simeq 0.29338. \tag{147}$$

The off diagonal elements are given by

$$C_0^{ss} = \frac{1}{4} \tag{148a}$$
$$C_1^{ss} \simeq 0.04269 \tag{148b}$$
$$C_2^{ss} = 0 \tag{148c}$$
$$C_0^{ts} = \frac{1}{2\sqrt{2\pi}} \tag{148d}$$
$$C_1^{ts} \simeq 0.1033 \tag{148e}$$
$$C_2^{ts} = 0. \tag{148f}$$

Here, $\simeq$ represents a numerical approximation of an integral. Notably, because both C_2^{ss} and C_2^{ts} are zero, the second-order term disappears from the approximation error. Thus, using $c_0 = (\delta_s + C_1^{ss})/C_0^{ss}$, the approximation error simplifies to

$$\epsilon_{apr} \approx \left(\frac{1}{2} - \frac{(C_0^{ts})^2}{C_0^{ss}} - \frac{(C_1^{ts})^2}{C_1^{ss}}\right) + \frac{(C_0^{ts})^2}{C_0^{ss}} \frac{c_0}{c_0 + L_h} + \frac{(C_1^{ts})^2}{2C_1^{ss}} \left(\sqrt{\left[\frac{L_h}{L_x} + \frac{\delta_s}{C_1^{ss}} - 1\right]^2 + \frac{4\delta_s}{C_1^{ss}}} - \left[\frac{L_h}{L_x} + \frac{\delta_s}{C_1^{ss}} - 1\right]\right). \tag{149}$$

7.3 Sparse ReLU

We can achieve sparse coding – which makes the model more relevant to experimental data – by shifting the threshold of ReLU: $g_s(u, b) \equiv \max(u - b, 0)$. The coefficients of the error are then given by

$$D_0^s = (1 + b^2)(1 - \Phi(b)) - \frac{b}{\sqrt{2\pi}} e^{-b^2/2} \tag{150a}$$
$$C_0^{ss} = \left(\frac{1}{\sqrt{2\pi}} e^{-b^2/2} - b(1 - \Phi(b))\right)^2 \tag{150b}$$
$$C_1^{ss} = (1 - \Phi(b))^2 \tag{150c}$$
$$C_2^{ss} = \frac{1}{4\pi} e^{-b^2} \tag{150d}$$

where $\Phi(b)$ is the cumulative Gaussian distribution: $\Phi(b) \equiv \int_{-\infty}^{b} ds \exp(-s^2/2)/\sqrt{2\pi}$.

7.4 Numerical estimation of the errors

The generalization error is easily estimated numerically by evaluating the test error over a large number of test samples,

$$\epsilon_{gen} \approx \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \left(\boldsymbol{w}_s \cdot g(\boldsymbol{J}_s \boldsymbol{x}_n) - y_n\right)^2. \tag{151}$$

In simulations of maximum likelihood learning, we calculated the weights using Eq. (51), then computed ϵ_{gen} using $N_{test} = 30,000$ samples. The cumulative generalization error under SGD learning was estimated using

$$\epsilon_{cg}^N \approx \frac{1}{N} \sum_{n=1}^{N} \left(\boldsymbol{w}_s^{(n-1)} \cdot g(\boldsymbol{J}_s \boldsymbol{x}_n) - y_n\right)^2. \tag{152}$$

Note that, because we provided a new sample $\{\boldsymbol{x}_n, y_n\}$ in each update, the right hand side is the cumulative test error, not the training error.

Estimating the approximation error (Eq. (12)), and the estimation error (Eq. (49)) from simulations is harder, becuase we need to evaluate $\langle \boldsymbol{g}_t \boldsymbol{g}_t^T \rangle$, $\langle \boldsymbol{g}_t \boldsymbol{g}_s^T \rangle$, and $\langle \boldsymbol{g}_s \boldsymbol{g}_s^T \rangle$ and the averages over $\boldsymbol{x}$ are generally intractable due to the high-dimensionality. However, if the nonlinearity, $g(\cdot)$, in both the teacher and student networks are ReLU, marginalization over $\boldsymbol{u} \equiv \boldsymbol{J}\boldsymbol{x}$ has a closed-form expression,

$$\begin{aligned} \langle g_q(u_i) g_{q'}(u_j) \rangle_{p(u_i, u_j)} &= \int_0^\infty du_i \int_0^\infty du_j \frac{u_i u_j}{2\pi \sigma_i \sigma_j \sqrt{1 - \rho_{ij}^2}} \exp\left(-\frac{1}{2(1 - \rho_{ij}^2)} \left[\frac{u_i^2}{\sigma_i^2} + \frac{u_j^2}{\sigma_j^2} - \frac{2\rho_{ij} u_i u_j}{\sigma_i \sigma_j}\right]\right) \\ &= \frac{\sigma_i \sigma_j}{2\pi} \left(\sqrt{1 - (\rho_{ij})^2} + \rho_{ij} \cos^{-1}(-\rho_{ij})\right) \end{aligned} \tag{153}$$

where $\sigma_i^2 = (\boldsymbol{J}_q \boldsymbol{J}_q^T)_{ii}$, $\rho_{ij} = \frac{(\boldsymbol{J}_q \boldsymbol{J}_{q'}^T)_{ij}}{\sigma_i \sigma_j}$, and the indices q and q' are either s or t. The errors under a specific network realization in Figs. 3, 5B, and 5C were calculated using this expression. For numerical stability, the inverse, $\boldsymbol{G}_s^{-1}$, was computed by solving a linear matrix equation $\boldsymbol{G}_s \boldsymbol{w}_s = \langle \boldsymbol{g}_s \boldsymbol{g}_t^T \rangle \boldsymbol{w}_t$, (see Eq. (8) and *github:nhiratani/olfactory_design*).

For the logistic activation, we computed (and plotted) only the generalization error, as numerical estimation of the approximation/estimation errors is difficult in this setting.

7.5 Model with evolutionary and developmental learning

In the model with low-precision hard-wired connections, the learning process is described as follows,

Initialize $\boldsymbol{J}_p^*$ by $J_{ij}^{P*} \sim N(0, 1/L_x)$;
for $m = 1, ..., 10^5$ **do**
  Update $\boldsymbol{w}_p^*$ and $\boldsymbol{J}_p^*$ using Eq. (139) and Eq. (138), alternatively.
end
Compress $\boldsymbol{w}_p^*$ and $\boldsymbol{J}_p^*$ into s_b bits weights $\boldsymbol{w}_p$ and $\boldsymbol{J}_p$ using Eq. (143);
Initialize $\boldsymbol{J}_s$ and $\boldsymbol{w}_s$ by $\boldsymbol{J}_s \sim N(0, 1/L_x)$ and $\boldsymbol{w}_s = \boldsymbol{0}$;
for $n = 1, ..., N$ **do**
  $\boldsymbol{x}_n \sim N(0, I)$, $y_n \sim N(\boldsymbol{w}_t \cdot g(\boldsymbol{J}_t \boldsymbol{x}_n), \sigma_t^2)$;
  $\boldsymbol{w}_s^{(n)} = \boldsymbol{w}_s^{(n-1)} + \frac{2}{\max(L_h, n)} (y_n - [\boldsymbol{w}_p \cdot \boldsymbol{g}(\boldsymbol{J}_p \boldsymbol{x}_n) + \boldsymbol{w}_s^{(n-1)} \cdot g(\boldsymbol{J}_s \boldsymbol{x}_n)]) g(\boldsymbol{J}_s \boldsymbol{x}_n)$;
end

In Fig. 6C we instead used Eq. (140) for the compression of the weights $\boldsymbol{J}_p^*$ and $\boldsymbol{w}_p^*$. In Fig. 6D, the compression was done with Eq. (143), and $\boldsymbol{w}_p$ was additionally trained in the developmental learning phase using

$$\boldsymbol{w}_p^{(n)} = \boldsymbol{w}_p^{(n-1)} + \frac{2}{\max(L_h, n)} (y_n - [\boldsymbol{w}_p^{(n-1)} \cdot g(\boldsymbol{J}_p \boldsymbol{x}_n) + \boldsymbol{w}_s^{(n-1)} \cdot g(\boldsymbol{J}_s \boldsymbol{x}_n)]) g(\boldsymbol{J}_p \boldsymbol{x}_n) \tag{154}$$

from the low-precision weight $\boldsymbol{w}_p^{(n=0)}$ derived from Eq. (143).

7.6 Model with low-dimensional structure in the input

So far we have assumed that the activity at the glomeruli, $\boldsymbol{x}$, follows an independent Gaussian distribution (see Eq. (6)). However, in the mammalian olfactory system there are at least twice as many glomeruli as receptor types [1]. In this regime the input is lower dimensional than the number of glomeruli, invalidating the assumption that the input follows an independent Gaussian distribution. To understand how low dimensional input affects the optimal hidden layer size, we investigated a model in which the input to the olfactory bulb had fixed dimension while the number of glomeruli was allowed to grow, and asked how the optimal hidden layer size depended on the number of glomeruli.

We let the input to the circuit have dimension L_z, and sample that input from an independent Gaussian distribution: $\boldsymbol{z} \in \Re^{L_z} \sim \mathcal{N}(0, \boldsymbol{I})$. The elements of $\boldsymbol{z}$ roughly correspond to the population activity of olfactory sensory neurons expressing one olfactory receptor gene. For simplicity, we assume that L_x is a multiple of L_z, and we use κ to denote that multiple (so $\kappa \equiv L_x / L_z$). We also assume that each receptor type projects to exactly κ glomeruli; combining this with the experimental observations that each glomerulus receives inputs from only one receptor type [2], we see that the $L_x \times L_z$ matrix, denoted $\boldsymbol{W}_z$, that transforms the input to the output must have the form

$$W_{z,ij} = \begin{cases} 1 & \kappa j - \kappa < i \leq \kappa j \\ 0 & \text{otherwise.} \end{cases} \tag{155}$$

In words: if, for instance, $\kappa = 5$, then the first olfactory receptor type will project to glomeruli 1-5, the second to glomeruli 6-10, and so on.

If the transformation from $\boldsymbol{z}$ to $\boldsymbol{x}$ were linear, activity would be constained to a linear L_z dimensional subspace, and adding glomeruli would have no effect on generalization error. However, the olfactory circuitry contains nonlinearities and lateral inhibition that may increase the linear dimensionality [3]. To reflect these factors, we use the same student model as before, $\hat{y} = \boldsymbol{w}_s \cdot g(J_s \boldsymbol{x})$ (Eq. (3)), but now with $\boldsymbol{x}$ given by

$$\boldsymbol{x} = \boldsymbol{W}_I^{-1} \left[g(W_z \boldsymbol{z} + \boldsymbol{b}_x) - \boldsymbol{m}_I \right] . \tag{156}$$

where $g(\cdot)$ is a ReLU nonlinearity, $\boldsymbol{b}_x$ is the bias, and $\boldsymbol{W}_I$ and $\boldsymbol{m}_I$ control the degree of lateral inhibition. The bias, $\boldsymbol{b}_x$, was set to

$$b_{x,i} = \Psi^{-1} \left(\frac{(i-1)\%\kappa + 1}{\kappa + 1} \right) \tag{157}$$

where % denotes mod and $\Psi(x)$ is the Gaussian cumulative distribution function. This ensures that glomeruli receiving input from the same receptor type experience a different nonlinearity. For the lateral inhibtion, we assume that $\boldsymbol{x}$ obeys the dynamics

$$\tau_I \dot{\boldsymbol{x}} = -W_I \boldsymbol{x} + g(W_z \boldsymbol{z} + \boldsymbol{b}_z) - \boldsymbol{m}_I \,, \tag{158}$$

with τ_I small, for which the fixed point satisfies Eq. (156). We set W_I and $\boldsymbol{m}_I$ empirically to

$$\boldsymbol{m}_I = \frac{1}{N_I} \sum_{t=1}^{N_I} g(W_z \boldsymbol{z}_t + \boldsymbol{b}_x), \tag{159a}$$

$$W_I = \left(\frac{1}{N_I} \sum_{t=1}^{N_I} \left(g(W_z \boldsymbol{z}_t + \boldsymbol{b}_x) - \boldsymbol{m}_I \right) \left(g(W_z \boldsymbol{z}_t + \boldsymbol{b}_x) - \boldsymbol{m}_I \right)^T \right)^{1/2} , \tag{159b}$$

with $N_I = 30000$. These weights can easily be learned with unsupervised Hebbian-type plasticity [30]. With this choice, the distribution of $\boldsymbol{x}$ will correspond, at least approximately, to independent white noise.

The teacher model was

$$y = \boldsymbol{w}_t \cdot g(\boldsymbol{J}_t \boldsymbol{z}) + \sigma_t \xi, \tag{160}$$

where $\boldsymbol{J}_t$ is an $L_t \times L_z$ random Gaussian matrix with variance $1/L_z$, $g(\cdot)$ is an element-wise nonlinearity, and $\sigma_t \xi$ is the teacher noise. This teacher model is the same as the original teacher model if $L_z = L_x$

In this setting, we estimated the optimal hidden layer size at different L_x, with L_z fixed, using maximum likelihood estimation (MLE). If the hidden layer size is determined purely by the number of olfactory receptor genes, then the optimal hidden layer size will be independent of L_x. In the absence of lateral inhibition, indeed the optimal hidden layer size shows very weak dependence on L_x (blue lines in Fig. S3; here we used the least-square method instead of MLE to estimate $\boldsymbol{w}$ because the covariance matrix often becomes singular). However, with whitening via lateral inhibition (that is, the model described above), the optimal hidden layer size exhibits approximately the same dependence on L_x as the model without any low-dimensional structure (orange vs gray lines in Fig. S3; gray line is the analytical estimation for $\boldsymbol{x} \sim N(0, I)$). These results were robust with respect to L_z (L_z was set to 250, 500, and 1000 in the left, middle, and right panel of Fig. S3). Thus, even if the input has an intrinsic low-dimensional structure, as long as there is a nonlinearity and lateral inhibition we see a similar scaling as the model with independent Gaussian input, as used in the main text.

7.7 Numerical estimation of the optimal hidden layer size

In both maximum likelihood and SGD simulations, we first estimated the generalization error by calculating the mean error over K_{sim} simulations, for various L_h spanning from $L_h = 10$ to $L_h = L_h^{\max}$ with a 10% increment at each step. We defined the empirical estimate of the optimal hidden layer size as the network size that yielded the minimum average error.

In the MLE simulations, we used $L_h^{\max} = \min[N, 30,000]$, except for the large L_x simulations in Fig. 4A, where we used $L_h^{\max} = 15,000$ for $L_x > 10,000$, $L_h^{\max} = 6,000$ for $L_x > 30,000$, and in Fig. 4E, where we set $L_h^{\max} = 4,000$. For each L_x, we took the mean over $K_{\text{sim}} = 100$ if $N < 1000$, else $K_{\text{sim}} = 10$.

In the SGD simulations, we set $L_h^{\max} = 30,000$ and $K_{\text{sim}} = 10$, except for Fig. 5B where we used $K_{\text{sim}} = 100$, and for the large L_x region of Fig. 4D, where we set $L_h^{\max} = 10,000$ for $L_x > 7,000$, and $L_h^{\max} = 3,300$ for $L_x > 20,000$). In Fig. 5F and Fig. 6, we used $L_h^{\max} = 100,000$.

8 Eigenvectors and eigenvalues of $\boldsymbol{G}_s$

Here we estimate the eigenvectors and eigenvalues of $\boldsymbol{G}_s$ using the approximate expression given in Eq. (30). That expression consists of four matrices: the identity, a rank one matrix with eigenvalue that scales as L_h, and, as pointed out immediately after Eq. (30), two matrices with Marchenko-Pastur distributions for their eigenvalues,

$$\boldsymbol{J}_s \boldsymbol{J}_s^T \text{:} \;\; \lambda \sim MP(1, L_h/L_x) \tag{161a}$$

$$\boldsymbol{M}_s \boldsymbol{M}_s^T \text{:} \;\; \lambda \sim MP(1, 2L_h/L_x^2) \,. \tag{161b}$$

For these matrices, so long as $L_h > L_x^2/2$, the nonzero eigenvalues scale as L_h/L_x and $2L_h/L_x^2$, respectively. In this regime, the nonzero eigenvalues of the three non-identity matrices in Eq. (30) are successively smaller, each time by a factor of L_x. We will assume this holds in general; when it does not, our approximation may not be very accurate.

To make use of the successively smaller eigenvalues, we note that if we sum two matrices with very different eigenvalues, the one with large eigenvalues dominates. More formally, consider two symmetric matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$, such that their nonzero eigenvalues are both $\mathcal{O}(1)$. Letting $\boldsymbol{v}_Q$ be an eigenvector of $\boldsymbol{Q}$ with eigenvalue λ_Q, for $|\epsilon| \ll 1$, we have

$$(\boldsymbol{Q} + \epsilon \boldsymbol{R})\boldsymbol{v}_Q = \boldsymbol{Q}\boldsymbol{v}_Q + \epsilon \boldsymbol{R}\boldsymbol{v}_Q = \lambda_Q \boldsymbol{v}_Q + \epsilon \boldsymbol{R}\boldsymbol{v}_Q \approx \lambda_Q \boldsymbol{v}_Q \,. \tag{162}$$

If $\boldsymbol{Q}$ is rank-deficient, there will be additional $\mathcal{O}(\epsilon)$ eigenvalues. Their eigenvectors will lie in the space spanned by $\boldsymbol{R}$, but with the space spanned by $\boldsymbol{Q}$ projected out.

We will now apply this to $\boldsymbol{G}_s$, but with a small correction, which turns out to be necessary to get good agreement with simulations: when computing the eigenvalue associated with the rank one matrix $\mathbf{1}_h \mathbf{1}_h^T$, we treat $\boldsymbol{J}_s \boldsymbol{J}_s^T$ and $\boldsymbol{M}_s \boldsymbol{M}_s^T$ as identity matrices, and when computing the eigenvalue spectrum associated with $\boldsymbol{J}_s \boldsymbol{J}_s^T$ we treat $\boldsymbol{M}_s \boldsymbol{M}_s^T$ as the identity matrix. (Note that $\boldsymbol{J}_s \boldsymbol{J}_s^T$ and $\boldsymbol{M}_s \boldsymbol{M}_s^T$ are typically rank deficient. However, we can consider an ensemble average; because their eigenvalues average to 1, that ensemble average is the identity matrix.)

Using this procedure, the relevant eigenvalue equation associated with the matrix associated with $\mathbf{1}_h \mathbf{1}_h^T$ is

$$\left((\delta_s + C_1^{ss} + C_2^{ss})\boldsymbol{I} + C_0^{ss} \mathbf{1}_h \mathbf{1}_h^T\right) \cdot \boldsymbol{v}^{(0)} = \lambda^{(0)} \boldsymbol{v}^{(0)} \,, \tag{163}$$

implying that

$$\lambda^{(0)} = \delta_s + C_1^{ss} + C_2^{ss} + C_0^{ss} L_h \tag{164a}$$

$$\boldsymbol{v}^{(0)} = \frac{\mathbf{1}_h}{\sqrt{L_h}} \,. \tag{164b}$$

To find the eigenvalues associated with $\boldsymbol{J}_s \boldsymbol{J}_s^T$, we should project out the one dimensional subspace spanned by $\mathbf{1}_h$, but that will have an $\mathcal{O}(1/L_h)$ effect, so we do not do it. Consequently, the relevant eigenvalue equation associated with the matrix $\boldsymbol{J}_s \boldsymbol{J}_s^T$ is

$$\left((\delta_s + C_2^{ss})\boldsymbol{I} + C_1^{ss} \boldsymbol{J}_s \boldsymbol{J}_s^T\right) \boldsymbol{v}_k^{(1)} = \lambda_k^{(1)} \boldsymbol{v}_k^{(1)} \,, \tag{165}$$

implying that

$$\lambda_k^{(1)} = \delta_s + C_2^{ss} + C_1^{ss} \tilde{\lambda}_k^{(1)} \tag{166}$$

where

$$\tilde{\lambda}^{(1)} \sim MP^+\left(1, \frac{L_h}{L_x}\right) \,. \tag{167}$$

The $+$ superscript on MP indicates that we should include only the non-zero eigenvalues.

To find the eigenvalues associated with $\boldsymbol{M}_s \boldsymbol{M}_s^T$, we need to project out the subsapce spanned by $\boldsymbol{J}_s \boldsymbol{J}_s^T$. Using $\widetilde{\boldsymbol{M}}_s$ to denote $\boldsymbol{M}_s$ in the lower dimensional space, the relevant eigenvalue equation is

$$\left(\delta_s \boldsymbol{I} + C_2^{ss} \widetilde{\boldsymbol{M}}_s \widetilde{\boldsymbol{M}}_s^T\right) \boldsymbol{v}_k^{(2)} = \lambda_k^{(2)} \boldsymbol{v}_k^{(2)} \,, \tag{168}$$

The dimension of the subspace we project out is $\max[L_h, L_x]$. Assuming that the projection $\boldsymbol{M} \to \widetilde{\boldsymbol{M}}$ is random, the eigenvalues of $\boldsymbol{M}_s \boldsymbol{M}_s^T$ are reduced by a factor of $[1 - L_x/L_h]^+$, giving us

$$\lambda_k^{(2)} = \delta_s + C_2^{ss} \tilde{\lambda}_k^{(2)} \tag{169}$$

where

$$\tilde{\lambda}^{(2)} \sim MP^+\left(\left[1 - \frac{L_x}{L_h}\right]^+, \frac{2L_h}{L_x^2}\right) \,. \tag{170}$$

Finally, if $L_r > 0$, there are additional eigenvectors. We have already taken care of the matrices with structure, so the remaining matrix is just $\delta_s \boldsymbol{I}$. Consequantly,

$$\lambda_k^{(r)} = \delta_s \,. \tag{171}$$

Because we need them for the analysis of SGD, we compute the average eigenvalues for the two components: $\tilde{\lambda}^{(1)}$ and $\tilde{\lambda}^{(2)}$. For the full Marchenko-Pastur distribution with parameters σ^2 and λ, the average eigenvalue is σ^2. However, for the distribution over only the non-zero eigenvalues, the average eigenvalue is $\sigma^2 \max[1, \lambda]$. Thus,

$$\lambda^{(0)} = \delta_s + C_1^{ss} + C_2^{ss} + C_0^{ss} L_h \tag{172a}$$

$$\langle \lambda_k^{(1)} \rangle = \delta_s + C_2^{ss} + C_1^{ss} \max\left[1, \frac{L_h}{L_x}\right] \tag{172b}$$

$$\langle \lambda_k^{(2)} \rangle = \delta_s + C_2^{ss} \left[1 - \frac{L_x}{L_h}\right]^+ \max\left[1, \frac{2L_h}{L_x^2}\right] \tag{172c}$$

$$\langle \lambda_k^{(r)} \rangle = \delta_s \tag{172d}$$

where we included $\lambda^{(0)}$(Eq. (164a)), and λ_r for completeness.

To compute the approximation error, we also need the eigenvalue/eigenvector expansion of the right hand side of Eq. (41). Repeating the above analysis, we find that

$$\lambda_k^{ts(0)} = \frac{(C_1^{ts})^2}{L_x} + \frac{(C_2^{ts})^2}{L_x^2/2} + (C_0^{ts})^2 L_h \approx (C_0^{ts})^2 L_h \tag{173a}$$

$$\lambda_k^{ts(1)} = \frac{(C_2^{ts})^2}{L_x^2/2} + \frac{(C_1^{ts})^2}{L_x} \tilde{\lambda}_k^{(1)} \approx \frac{(C_1^{ts})^2}{L_x} \tilde{\lambda}_k^{(1)} \tag{173b}$$

$$\lambda_k^{ts(2)} = \frac{(C_2^{ts})^2}{L_x^2/2} \tilde{\lambda}_k^{(2)} \tag{173c}$$

$$\lambda_k^{ts(r)} = 0 \tag{173d}$$

where the approximations are valid in the large L_x limit.

9 Marchenko-Pastur averages

In §3 (see in particular Eq. (42)) we need to compute averages of the form

$$\frac{1}{L} \sum_{k=1}^{L'} \frac{\lambda_k}{c + \lambda_k} = \frac{L'}{L} \left\langle \frac{\lambda}{c + \lambda} \right\rangle_{\lambda \sim MP^+(\sigma^2, \bar{\lambda})} \tag{174}$$

where, as above the $+$ superscript on MP indicates that the average is over only the positive eigenvalues. Computing analytically the average on the right hand side, we have

$$\frac{L'}{L} \left\langle \frac{\lambda}{c + \lambda} \right\rangle_{\lambda \sim MP^+(\sigma^2, \bar{\lambda})} = \frac{L'/L}{\min\left[1, \bar{\lambda}\right]} \left(1 - f\left(\bar{\lambda}; \frac{c}{\sigma^2}\right)\right) \tag{175}$$

where

$$f(\bar{\lambda}; c) \equiv \frac{\sqrt{\left(\bar{\lambda} - 1 + c\right)^2 + 4c} - \left(\bar{\lambda} - 1 + c\right)}{2} \,. \tag{176}$$

The large and small $\bar{\lambda}$ limits of f are relatively simple,

$$f(\bar{\lambda}, c/\sigma^2) \to \begin{cases} 1 & \bar{\lambda} \to 0 \\ c/(\sigma^2 \bar{\lambda}) & \bar{\lambda} \to \infty \,. \end{cases} \tag{177}$$

The small $\bar{\lambda}$ limit is important, because it tells us that $1 - f(\bar{\lambda}, c/\sigma^2)$ is small whenever $\bar{\lambda}$ is small.

For the first sum in Eq. (42), $L = L_x$, $L' = L_1 = \min[L_x, L_h]$, and $c = c_1$ (defined in Eq. (36b)). The parameters of the Marchenko-Pastur distribution, given in Eq. (167), are $\sigma^2 = 1$ and $\bar{\lambda} = L_h/L_x$. The latter implies that $L'/(L \min[1, \bar{\lambda}]) = 1$. Consequently, the first sum in Eq. (42) is

$$\frac{1}{L_x} \sum_{k=1}^{L_1} \frac{\tilde{\lambda}_k^{(1)}}{c_1 + \tilde{\lambda}_k^{(1)}} = 1 - f(\bar{\lambda}; c_1) \,. \tag{178}$$

For the second sum in Eq. (42), $L = L_x^2/2$, $L' = L_2 = \min[L_x^2/2, L_h - L_x]^+$ and $c = c_2$ (defined in Eq. (36c). The parameters of the Marchenko-Pastur distribution, given in Eq. (170), are $\sigma^2 = [1 - L_x/L_h]^+$ and $\bar{\lambda} = 2L_h/L_x^2$. We thus have, after a small amount of algebra,

$$\frac{1}{L_x^2/2}\sum_{k=1}^{L_2}\frac{\tilde{\lambda}_k^{(2)}}{c_2+\tilde{\lambda}_k^{(2)}} = \frac{\min[L_x^2/2, L_h - L_x]^+}{\min[L_x^2/2, L_h]^+}\left(1 - f\left(\frac{L_h}{L_x^2/2}; \frac{c_2}{[1-L_x/L_h]^+}\right)\right) . \tag{179}$$

Noticing that the first term is 1 at $L_h > L_x^2/2 + L_x$, $[1-L_x/L_h]^+$ at $L_h < L_x^2/2$, and slightly smaller than 1 in between, we can simplify the expression above as

$$\frac{1}{L_x^2/2}\sum_{k=1}^{L_2}\frac{\tilde{\lambda}_k^{(2)}}{c_2+\tilde{\lambda}_k^{(2)}} \approx \left[1 - \frac{L_x}{L_h}\right]^+\left[1 - f\left(\frac{L_h}{L_x^2/2}; \frac{c_2}{1-L_x/L_h}\right)\right] . \tag{180}$$

In §4.1 we need the average of the inverse of the eigenvalue. That is easily found from the above analysis,

$$\left\langle\frac{1}{\lambda}\right\rangle_{\lambda\sim MP^+(\sigma^2,\bar{\lambda})} = \left.\frac{\partial}{\partial c}\right|_{c=0}\left\langle\frac{-\lambda}{c+\lambda}\right\rangle_{\lambda\sim MP^+(\sigma^2,\bar{\lambda})} . \tag{181}$$

Using Eq. (175) for the right hand side, a straightforward calculation yields

$$\left\langle\frac{1}{\lambda}\right\rangle_{\lambda\sim MP^+(\sigma^2,\bar{\lambda})} = \frac{1}{\sigma^2|\bar{\lambda}-1|} . \tag{182}$$

References

[1] Shyam Srinivasan and Charles F Stevens. Scaling principles of distributed circuits. *Current Biology*, 29(15):2533–2540, 2019.

[2] Helen B Treloar, Paul Feinstein, Peter Mombaerts, and Charles A Greer. Specificity of glomerular targeting by olfactory sensory axons. *Journal of Neuroscience*, 22(7):2469–2477, 2002.

[3] Rachel I Wilson and Zachary F Mainen. Early events in olfactory processing. *Annu. Rev. Neurosci.*, 29:163–201, 2006.

[4] KD Ernst, J Boeckh, and V Boeckh. A neuroanatomical study on the organization of the central antennal pathways in insects. *Cell and tissue research*, 176(3):285–308, 1977.

[5] Qian Gao, Bingbing Yuan, and Andrew Chess. Convergent projections of drosophila olfactory neurons to specific glomeruli in the antennal lobe. *Nature neuroscience*, 3(8):780, 2000.

[6] Joshua P Martin, Aaron Beyerlein, Andrew M Dacks, Carolina E Reisenman, Jeffrey A Riffell, Hong Lei, and John G Hildebrand. The neurobiology of insect olfaction: sensory processing in a comparative context. *Progress in neurobiology*, 95(3):427–447, 2011.

[7] Joachim Schachtner, Manfred Schmidt, and Uwe Homberg. Organization and evolutionary trends of primary olfactory brain centers in tetraconata (crustacea+ hexapoda). *Arthropod Structure & Development*, 34(3):257–299, 2005.

[8] Ariane Ramaekers, Edwige Magnenat, Elizabeth C Marin, Nanaë Gendre, Gregory SXE Jefferis, Liqun Luo, and Reinhard F Stocker. Glomerular maps without cellular redundancy at successive levels of the drosophila larval olfactory circuit. *Current biology*, 15(11):982–992, 2005.

[9] Liria M Masuda-Nakagawa, Nanaë Gendre, Cahir J O'Kane, and Reinhard F Stocker. Localized olfactory representation in mushroom bodies of drosophila larvae. *Proceedings of the National Academy of Sciences*, 106(25):10314–10319, 2009.

[10] Katharina Eichler, Feng Li, Ashok Litwin-Kumar, Youngser Park, Ingrid Andrade, Casey M Schneider-Mizell, Timo Saumweber, Annina Huser, Claire Eschbach, Bertram Gerber, et al. The complete connectome of a learning and memory centre in an insect brain. *Nature*, 548(7666):175, 2017.

[11] Sophie JC Caron, Vanessa Ruta, LF Abbott, and Richard Axel. Random convergence of olfactory inputs in the drosophila mushroom body. *Nature*, 497(7447):113, 2013.

[12] Yoshinori Aso, Daisuke Hattori, Yang Yu, Rebecca M Johnston, Nirmala A Iyer, Teri-TB Ngo, Heather Dionne, LF Abbott, Richard Axel, Hiromu Tanimoto, et al. The neuronal architecture of the mushroom body provides a logic for associative learning. *Elife*, 3:e04577, 2014.

[13] Sylvia Anton and Bill S Hansson. Central processing of sex pheromone, host odour, and oviposition deterrent information by interneurons in the antennal lobe of female spodoptera littoralis (lepidoptera: Noctuidae). *Journal of comparative neurology*, 350(2):199–214, 1994.

[14] Marcus Sjöholm, Irina Sinakevitch, Rickard Ignell, Nicholas J Strausfeld, and Bill S Hansson. Organization of kenyon cells in subdivisions of the mushroom bodies of a lepidopteran insect. *Journal of Comparative Neurology*, 491(3):290–304, 2005.

[15] Zhifeng Wang, Pengcheng Yang, Dafeng Chen, Feng Jiang, Yan Li, Xianhui Wang, and Le Kang. Identification and functional analysis of olfactory receptor family reveal unusual characteristics of the olfactory system in the migratory locust. *Cellular and Molecular Life Sciences*, 72(22):4429–4443, 2015.

[16] Beulah Leitch and Gilles Laurent. Gabaergic synapses in the antennal lobe and mushroom body of the locust olfactory system. *Journal of comparative Neurology*, 372(4):487–514, 1996.

[17] Gérard Arnold, Claudine Masson, and Sati Budharugsa. Comparative study of the antennal lobes and their afferent pathway in the worker bee and the drone (apis mellifera). *Cell and tissue research*, 242(3):593–605, 1985.

[18] Wolfgang Witthöft. Absolute anzahl und verteilung der zellen im him der honigbiene. *Zeitschrift für Morphologie der Tiere*, 61(1):160–184, 1967.

[19] Hidehiro Watanabe, Hiroshi Nishino, Michiko Nishikawa, Makoto Mizunami, and Fumio Yokohari. Complete mapping of glomeruli based on sensory nerve branching pattern in the primary olfactory center of the cockroach periplaneta americana. *Journal of Comparative Neurology*, 518(19):3907–3930, 2010.

[20] Sarah M Farris and Nicholas J Strausfeld. Development of laminar organization in the mushroom bodies of the cockroach: Kenyon cell proliferation, outgrowth, and maturation. *Journal of Comparative Neurology*, 439(3):331–351, 2001.

[21] Bertram Gerber and Reinhard F Stocker. The drosophila larva as a model for studying chemosensation and chemosensory learning: a review. *Chemical senses*, 32(1):65–89, 2006.

[22] EA Capaldi, GE Robinson, and SE Fahrbach. Neuroethology of spatial learning: the birds and the bees. *Annual review of psychology*, 50(1):651–682, 1999.

[23] Makoto Mizunami, Yukihisa Matsumoto, Hidehiro Watanabe, and Hiroshi Nishino. Olfactory and visual learning in cockroaches and crickets. In *Handbook of Behavioral Neuroscience*, volume 22, pages 549–560. Elsevier, 2013.

[24] Robi Tacutu, Thomas Craig, Arie Budovsky, Daniel Wuttke, Gilad Lehmann, Dmitri Taranukha, Joana Costa, Vadim E Fraifeld, and João Pedro De Magalhães. Human ageing genomic resources: integrated databases and tools for the biology and genetics of ageing. *Nucleic acids research*, 41(D1):D1027–D1033, 2012.

[25] Vladimir A Marυcenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

[26] Madhu Advani and Surya Ganguli. Statistical mechanics of optimal convex inference in high dimensions. *Physical Review X*, 6(3):031034, 2016.

[27] Mehmet Fişek and Rachel I Wilson. Stereotyped connectivity and computations in higher-order olfactory neurons. *Nature neuroscience*, 17(2):280, 2014.

[28] Peter Grunwald and Paul Vitányi. Shannon information and kolmogorov complexity. *arXiv preprint cs/0410002*, 2004.

[29] Geoffrey Hinton and Drew Van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*. Citeseer, 1993.

[30] Naoki Hiratani and Tomoki Fukai. Mixed signal learning by spike correlation propagation in feedback inhibitory circuits. *PLoS computational biology*, 11(4):e1004227, 2015.