Accepted 9-13-2024 in Journal of Computational Neuroscience

Computational model of layer 2/3 in mouse primary visual cortex explains observed visuomotor mismatch response

Heiko Hoffmann[1*]

[1*]Magimine, LLC, P.O. Box 941154, Simi Valley, 93094, CA, USA.

Corresponding author(s). E-mail(s): heiko@magimine.com;

Abstract

Activity in layer 2/3 of the mouse primary visual cortex has been shown to depend both on visual input and the mouse's locomotion. Moreover, this activity is altered by a mismatch between the observed visual flow and the predicted visual flow from locomotion. Here, I present a simple computational model that explains previously reported recordings from layer 2/3 neurons in mice. In my model, layer 2/3 encodes the velocity difference between the estimate from visual flow and the prediction from locomotion using a neural population code. Moreover, I describe a hypothesized mechanism for how the brain may carry out computations of variables encoded in population codes. This mechanism may point to a general principle for computing any mathematical function in the brain.

Keywords: Primary visual cortex, population code, visuomotor, prediction error

1 Introduction

The primary visual cortex, also known as V1, plays a fundamental role in early visual processing. It processes basic visual features such as orientation, color, and visual flow. In recent years, evidence has been presented that shows that neural activity in V1 also depends on the animal's motor output [Jordan and Keller, 2020, Keller et al, 2012, Leinweber et al, 2017, Miura and Scanziani, 2022]. This observation is consistent with the concept of internal models that predict the sensory consequences of motor commands [Hoffmann, 2007].

To explore the mechanism of these internal models, experiments have been carried out that create a mismatch between the actual sensory input and the sensory input that would be predicted given a certain motor output, e.g., [Jordan and Keller, 2020]. This mismatch can be realized using a virtual reality setup, where, e.g., a head-fixed mouse runs on a spherical treadmill, while receiving controlled visual input from a screen surrounding the mouse [Keller et al, 2012]. These experiments have shown that layer 2/3 in V1 responds to a mismatch between predicted and observed visual flow [Jordan and Keller, 2020]. But the neural mechanism behind this mismatch response is still being debated [Muzzu and Saleem, 2021, Vasilevskaya et al, 2023].

One mechanism that has been put forth is that the motor activity changes the gain on the V1 neurons [Fu et al, 2014]. This idea, however, has been challenged by experiments [Vasilevskaya et al, 2023]. A recently more prominent hypothesis is that the mismatch response corresponds to a prediction error [Hertäg and Clopath, 2022, Keller and Mrsic-Flogel, 2018, Mikulasch et al, 2023b], where single neurons represent this error. In contrast, population coding is ubiquitous in the visual cortex [Miura and Scanziani, 2022, Pouget et al, 2000, Tanabe, 2013]. In a population code, each neuron represents a preferred value and gets more active the closer the encoded variable approaches this preferred value [Hubel and Wiesel, 1959]. As a group, the activity of the neurons resembles a probability distribution of the encoded variable [Pouget et al, 2000].

Based on such population coding, I propose a simple computational model for explaining the mismatch response. The predictions from this model explain details observed by Jordan and Keller [2020] better than a prediction-error at the single-neuron level. The novelty of the model is to encode the prediction error in a *population* of layer 2/3 neurons.

In the following, I describe the computational model and its predictions of the mismatch response in more detail. The last part of the Results section describes how a biological neural network could compute a difference of variables that are encoded in population codes. Implementation details of the results are in Methods, which is followed by discussions. Finally, the article concludes by presenting ideas for future work.

2 Results

My computational model encodes the difference in velocity between the estimation from visual flow and the prediction from locomotion in a population of neurons. This velocity may refer to the mouse's body, with the difference representing a prediction error. Alternatively, the velocity difference may refer to other moving objects in a dynamic environment, as it is not explained by the mouse's self-motion.

The velocity is generally a vector. Here, I assume that groups of neurons encode one-dimensional projections of this vector. Together these groups could reconstruct the original velocity vector. This encoding is different from the traditional encoding hypothesis where neurons encode the direction of a vector [Hubel and Wiesel, 1959]. The advantage of the 1-d-projection encoding is that additions and subtractions of vectors can be reduced to additions and subtractions in each of the 1-d projection subspaces. Moreover, changes in speed, i.e., the magnitude of the velocity vector, will scale each projection by the same factor. So, to study the impact of locomotion speed, it is sufficient to model the population code for only a single 1-d projection, where each neuron has a preferred speed.

Here, I encoded the speed difference, Δv, between the estimation from visual observation, v_O, and the prediction from locomotion, v_L, in a population code of $N = 100$ neurons. Each neuron responds to a preferred value, uniformly distributed in the interval $[-1 - v_{\text{Off}}; 1 - v_{\text{Off}}]$, where v_{Off} offsets the range from being centered at 0 (Fig. 1). The unit of speed is arbitrary. The response of neurons in the model represents the average activity of neurons over many trials under the same conditions.

With this model, I studied two experimental conditions, as also described by Jordan and Keller [2020]. In the match condition, the visual-flow-based speed matched the prediction from locomotion. That is, our neural population encodes the difference, which is zero speed (Fig. 1A). In the mismatch condition, the observed visual-flow was zero, but the prediction from locomotion was non-zero. Here, the encoded speed difference is the negative of the locomotion speed.

The goal of my simulations is to predict the mismatch response (vertical arrows in Fig. 1A) between the two conditions as function of the locomotion speed and compare with results from neural recordings reported by Jordan and Keller [2020]. I simulated 10 trials, with locomotion speeds, v_L, from 0.05 to 0.5 in increments of 0.05. For each trial, I computed the population encoding for the match and mismatch conditions (Fig. 1A shows a trial for locomotion speed 0.5).

Same as Jordan and Keller [2020], I *operationally* define two types of neurons: depolarizing mismatch (dMM) neurons and hyperpolarizing mismatch (hMM) neurons. Among the N neurons, dMM neurons were those where the average activation across speed trials in the mismatch condition was at least 0.05 higher than in the match condition, and hMM neurons were those where the average activation in mismatch was at least 0.05 lower compared to match.

The resulting ratio of dMM and hMM neurons depended on the offset of the encoded speed range (Fig. 1B). For an offset of $v_{\text{Off}} = 0.76$, the ratio of dMM to hMM neurons matched best the mouse experiments [Jordan and Keller, 2020]: 51 to 18 neurons versus 17 to 6 in experiment. The remaining 31 neurons were unclassified. Using the

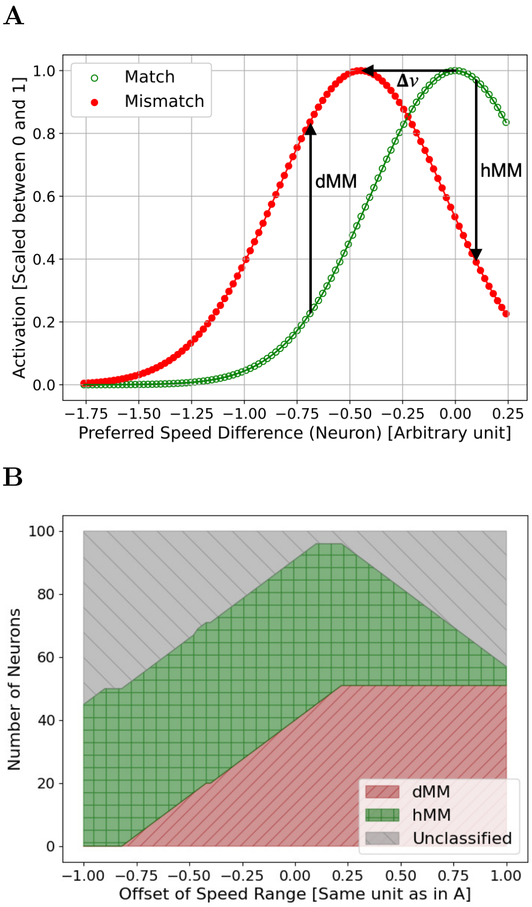


Fig. 1: Neural population encoding the speed difference between the predictions from visual flow and locomotion. (A) In the Match condition, the difference is zero. In the Mismatch condition, the visual flow is zero, resulting in the difference Δv. Depolarizing neurons (dMM) show increased activity, and hyperpolarizing neurons (hMM) show decreased activity. (B) The number of dMM and hMM neurons depends on the offset of the speed range (in A, $v_{\text{Off}} = 0.76$).

simulation results as ground truth, we can calculate the expected experimental numbers when picking 32 neurons (as in [Jordan and Keller, 2020]) at random out of our 100. The result is in good agreement with the experiment: n(dMM) $= 16.3 \pm 2.3$, n(hMM) $= 5.8 \pm 1.8$, and n(unclassified) $= 9.9 \pm 2.2$. Jordan and Keller [2020] reported 9 unclassified neurons.

Using the offset $v_{\text{Off}} = 0.76$, the resulting mismatch response positively correlated with locomotion speed for dMM neurons (linear fit with slope 0.95 ± 0.03) and negatively for hMM neurons (linear fit with slope -1.21 ± 0.05), as also observed by Jordan and Keller [2020]. The correlation coefficients are in good agreement with experiment, showing dMM neurons with a median above 0.5, hMM with a median below -0.5, and unclassified neurons with near zero correlation and a bias toward positive values (Fig. 2, compare with Figure 2G in Jordan and Keller [2020]).

This correlation can be understood by considering the shift in the tuning curve for the mismatch condition (Fig. 1A): the larger the shift, the larger the activation difference for most neurons, and therefore, the mismatch response. In contrast, for single-cell prediction errors, additional elements (e.g., inhibition) would be required to explain the sign difference in correlation between dMM and hMM neurons.

Our model encodes the prediction error or speed difference in a population code. Given that the observed and predicted speeds are likely encoded in population codes themselves, V1 would need to compute a subtraction on population-encoded variables. For this computation, I hypothesize the following mechanism. I assume both v_O and v_L are encoded in a neural population, and we want the resulting $\Delta v = v_O - v_L$ to be encoded in a neural population as well.

This computation can be achieved by logical AND operations in dendritic branches, as explained in the following example (Fig. 3). In code v_O, a neuron fires, which has a preferred value of a_5. Simultaneously, in code v_L, another neuron fires, which has a preferred value b_2. Both neurons connect to the same dendritic branch of a neuron in code $v_O - v_L$ that has a preferred value of c_4. Here, c_4 happens to match the difference $a_5 - b_2$. The branch and the corresponding neuron fire if and only if both incoming axons from a_5 and b_2 carry a spike (Fig. 3B).

In biology, there is evidence for such dendritic computation: spikes on nearby (20 μm) synaptic connections on a branch are added with a sigmoidal function [Polsky et al, 2004] (an AND operation is a simplified sigmoidal function: f(0+0) = 0, f(0+1) = 0, and f(1+1) = 1). Moreover, dendrites branch into multiple subdivisions

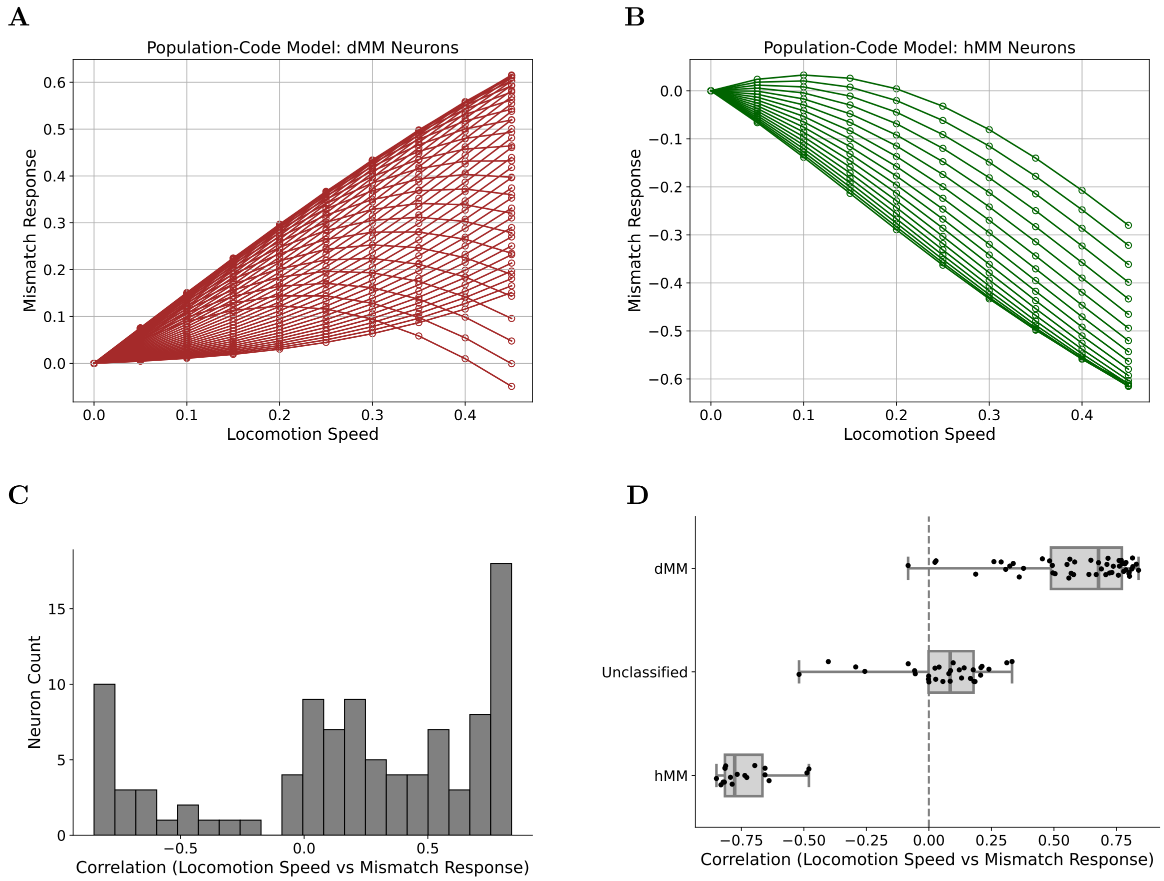


Fig. 2: Predicted dependence of mismatch response on locomotion speed, v_L, for dMM (A) and hMM (B) neurons. Each curve shows the response for a single neuron. Units are same as in Figure 1. (C) Correlation between mismatch response and locomotion across all population-code neurons. (D) Boxplot of correlation depending on neural classification. Here, $v_{\mathrm{Off}} = 0.76$ for all panels.

[Larkman and Mason, 1990], allowing a separate computation for each combination of neurons with preferred values a_i and b_j. Here, I assume neural connections for all possible combinations resulting in $c_k = a_i - b_j$ (Fig. 3A shows two of them). Hebbian learning might be one mechanism to create these connections between preferred values.

This mechanism ensures that any combination of firing neurons i and j will result in triggering a neuron k, where $c_k = a_i - b_j$. In turn, neuron k fires only if neurons with preferred values a_i and $b_j = a_i - c_k$ fire. On the neural level, there is no actual subtraction to encode the difference; computation happens through connecting neurons that encode appropriate preferred values.

I define the activity of neurons in a code as the average number of spikes fired over a given time window. If the activity of neurons in code v_O is described by the probability $p_{v_O}(a_i)$ and the activity in code v_L by $p_{v_L}(b_j)$, then, given our mechanism, the resulting code for $v_O - v_L$ has the distribution

$$p(c_k) = \sum_i p_{v_O}(a_i) p_{v_L}(a_i - c_k)\,. \qquad (1)$$

In the continuous limit for large N, we obtain

$$p(\Delta v) = \int p_{v_O}(x) p_{v_L}(x - \Delta v) dx\,. \qquad (2)$$

This distribution is our desired population code for the difference $\Delta v = v_O - v_L$. Moreover, $p(\Delta v)$

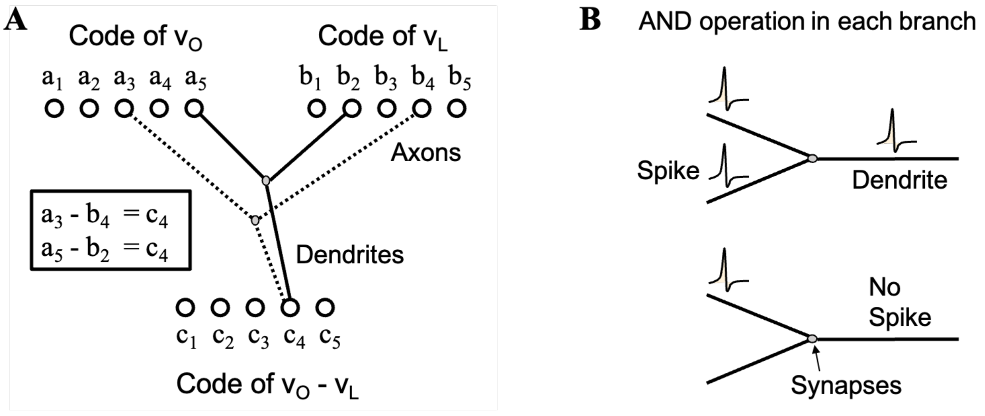


Fig. 3: Hypothesized subtraction of variables encoded in population codes. (A) Each dendritic branch represents a difference between two specific values of the speed variables v_O and v_L. (B) A branch carries out an AND operation on the receiving spikes.

is computed as expected from Bayesian statistics given uncertainty about the values of v_O and v_L, as expressed in the probability distributions $p_{v_O}(x)$ and $p_{v_L}(y)$.

3 Methods

To compute the activation A_i of neuron i, I used a Gaussian function of the encoded speed difference, Δv,

$$A_i = \exp\left(-(\Delta v - \mu_i)^2/(2\sigma^2)\right), \tag{3}$$

where μ_i is the preferred speed difference of neuron i and $\sigma = 0.4$.

The mismatch response, ΔA_i, is the difference between the activation in the mismatch and the match condition,

$$\Delta A_i = A_{i,\text{mismatch}} - A_{i,\text{match}} \,. \tag{4}$$

To compute the expected number of dMM and hMM neurons, I computed, first, the probability, p, of finding n_d dMM, n_h hMM, and n_u unclassified neurons when selecting n neurons at random out of N,

$$p = \frac{\binom{g_d}{n_d}\binom{g_h}{n_h}\binom{g_u}{n_u}}{\binom{N}{n}}, \tag{5}$$

where g_d, g_h, and g_u are the ground truth numbers from our simulation ($g_d = 51$, $g_h = 18$, and $g_u = 31$ for $N = 100$). This probability was computed for all possible values of n_d, n_h, and n_u, allowing evaluating the expected value and SD for each of these variables.

The reported slopes in the text for the responses of dMM and hMM neurons were computed using linear fits to ΔA versus v_L across all neurons and trials for each neuron class. The linear fits used bisquare weighting of residuals.

For each neuron i, I computed the Pearson correlation coefficient between response and speed,

$$r_i = \frac{\sum_{t=1}^{T}(\Delta A_i^t - \overline{\Delta A_i})(v_L^t - \overline{v_L})}{\sqrt{\sum_{t=1}^{T}(\Delta A_i^t - \overline{\Delta A_i})^2}\sqrt{\sum_{t=1}^{T}(v_L^t - \overline{v_L})^2}}, \tag{6}$$

where t is the trial index and T the total number of trials. Before computing the correlation, I reproduced each speed trial 20 times with different mean-zero Gaussian noise (SD = 0.15) added to the response.

4 Discussion

A simple model of population coding the difference in predicted and observed speed explained multiple phenomena observed in mouse experiments: the ratio of dMM, hMM, and unclassified neurons, the positive correlation of dMM response with locomotion speed, and the negative correlation of hMM response with locomotion speed. Those phenomena are a consequence of the population coding. A single-cell prediction error could not predict these phenomena without additional model elements, like, e.g., inhibitory input to hMM neurons to explain the different sign of correlation.

The difference in predicted and observed speed is commonly interpreted as a prediction error [Hertäg and Clopath, 2022, Keller and Mrsic-Flogel, 2018, Mikulasch et al, 2023b]. But this difference may be also interpreted as the optical-flow component not explained by self motion [Mikulasch et al, 2023a]. My population-code model is agnostic to the interpretation and to how the brain might use the encoded information. Moreover, this model is agnostic to the neural dynamics that will result in the average neural activity as represented by our population code.

As Jordan and Keller [2020], I defined dMM and hMM neurons operationally. As such, there may be no physiological difference between these neurons, and as my model shows, there does not need to be any. Moreover, the model explains why there are unclassified neurons: they are part of the population code, but their change in response under mismatch just happened to be small, e.g., because the preferred value of a neuron is far away from the encoded speed and due to the Gaussian tuning the resulting changes in activation will be small.

In our model, the ratio of dMM and hMM neurons varied with the range of the encoded speed values. The range offset $v_{\mathrm{Off}} = 0.76$ gave the best match of the dMM/hMM ratio with experiment. Coincidently, for this offset, the resulting mismatch response also agreed better with the experiment compared to other offsets: Figure S1 in the Supplementary information shows mismatch responses for different offset values. While the mismatch response was stable for a range of values, for much smaller and negative offsets, the response distribution drifted away from the experimental results.

A positive offset implies that the population code can encode a larger range of negative prediction errors, where v_L is larger than v_O. This bias is reasonable if those values are more likely, as indeed appears to be the case: a speed prediction from locomotion can be larger than visually estimated, e.g., when the feet are sliding on the ground, while the optical flow stays constant.

For the results, I used a fixed number of neurons and speed trials. Varying these numbers just changes the density of the data points in the mismatch-response scatter plots (Fig. 2A and B): more speed trials lead to a higher density in the x direction and more neurons to a higher density in the y direction.

Moreover, for simplicity, I used a uniform spacing of preferred values in a population code. Changing this setting to randomly distributed preferred values had no significant impact on the results. The curves in Fig. 2A and B would merely be less regularly spaced. (The supplementary gitlab code provides an option to test the random distribution variant.)

My threshold for defining dMM and hMM neurons was about 10% of the maximum mismatch response, which is similar to the value used by Jordan and Keller [2020]. Increasing this threshold naturally reduces the number of dMM and hMM neurons. In addition, a larger threshold resulted in mismatch responses with a smaller variance, following closer the linear fit, while maintaining the sign of the slope for both dMM and hMM neurons.

For simplicity, I focused on Gaussian tuning curves. In the Supplementary information, Fig. S2 shows results for cosine tuning curves, having two peaks. In that case, the dMM response still positively correlated with locomotion speed and the hMM response negatively. However, for any combination of offset and threshold, the ratios of dMM, hMM, and unclassified neurons deviated from the mouse experiments. So, Gaussian tuning agreed better with experiment.

Another simplification was using only one dimension of the velocity, but when recording from the mouse, we expect a mix of neurons from population codes for different velocity directions. This increased complexity, however, had only a minimal impact on the mismatch response (Fig. S4 in Supplementary information).

While I provided arguments that layer 2/3 in V1 encodes the prediction error in a population code, I do not suggest that all of layer 2/3 neurons participate in this encoding. It may be just one of the functions of this layer.

The mechanism of how the cortex carries out mathematical operations on population-encoded variables is still unknown. I provided a hypothesis for generating a population code that represents the difference of two population encoded-variables. This mechanism probabilistically integrates two variables with given uncertainties expressed in probability distributions. It may therefore provide a path for Bayesian estimation in the brain [Körding and Wolpert, 2004].

This new concept for computing the difference between two population-encoded variables is not limited to subtractions, but any mathematical function could be computed. For example, to compute $f(a, x, y) = a \sin(x + y)$, we connect three neurons with specific preferred a, x, and y values through one dendritic branch that belongs to a neuron with preferred value $a \sin(x + y)$. The connections could be created with Hebbian learning, and the nervous system learns functional mappings without explicitly representing any function.

5 Conclusion

This article demonstrated that a population code of the prediction error, the difference between visual-flow-based velocity estimate and its prediction from locomotion, could explain details observed in neural recordings from layer 2/3 in V1 of mice. Despite its simplicity, our model could predict ratios of dMM, hMM, and unclassified neurons and the correlations of their mismatch responses with locomotion speed. In contrast, the single-cell prediction-error hypothesis requires more explanations, like different connections to inhibitory neurons between dMM and hMM neurons.

The computation of variables encoded in population codes could be a major function of the cortex, and I suggested a plausible mechanism for carrying out such computations. Moreover, connecting neurons via multiple dendritic branches that carry out local computations provides a new perspective for creating artificial neural networks that hasn't been exploited yet, despite its apparent applications in biology.

Our future work aims at developing new computational models based on these principles and testing these models on experimental data. For example, we plan to uncover the nature of how the brain deals probabilistically with sensory uncertainty. I hope that this work will lead to a theory of the cortex, making testable quantitative predictions, akin to the field of physics.

Supplementary information. The supplementary information shows three additional results: first, the mismatch responses for different velocity ranges encoded by the population code, second, the mismatch responses for a population code with bimodal tuning curves, and third, the mismatch responses when recording from neurons selected from three different population codes, each encoding the projection of a 2-dimensional velocity vector onto a basis vector that is different for each population code.

The code to run my computational model and recreate the results is available at https://github.com/drawfind/visuomotor_mismatch.

Acknowledgments. The author thanks Richard Hoffmann for carefully reviewing the manuscript.

Statements and Declarations

No funding was received for conducting this study. The author has no relevant financial or non-financial interests to disclose.

References

Fu Y, Tucciarone J, Espinosa J, et al (2014) A cortical circuit for gain control by behavioral state. Cell 156(6):1139–1152

Hertäg L, Clopath C (2022) Prediction-error neurons in circuits with multiple neuron types: Formation, refinement, and functional implications. Proceedings of the National Academy of Sciences 119(13):e2115699119

Hoffmann H (2007) Perception through visuomotor anticipation in a mobile robot. Neural Networks 20(1):22–33

Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate cortex. J Physiol 148(3):574–591

Jordan R, Keller GB (2020) Opposing influence of top-down and bottom-up input on excitatory layer 2/3 neurons in mouse primary visual cortex. Neuron 108(6):1194–1206.e5

Keller G, Bonhoeffer T, Hübener M (2012) Sensorimotor mismatch signals in primary visual cortex of the behaving mouse. Neuron 74(5):809–815

Keller GB, Mrsic-Flogel TD (2018) Predictive processing: A canonical cortical computation. Neuron 100(2):424–435

Körding KP, Wolpert DM (2004) Bayesian integration in sensorimotor learning. Nature 427(6971):244–247

Larkman A, Mason A (1990) Correlations between morphology and electrophysiology of pyramidal neurons in slices of rat visual cortex. i. establishment of cell classes. J Neurosci 10(5):1407–1414

Leinweber M, Ward DR, Sobczak JM, et al (2017) A sensorimotor circuit in mouse cortex for visual flow predictions. Neuron 95(6):1420–1432.e5

Mikulasch FA, Rudelt L, Priesemann V (2023a) Visuomotor mismatch responses as a hallmark of explaining away in causal inference. Neural computation 35(1):27–37

Mikulasch FA, Rudelt L, Wibral M, et al (2023b) Where is the error? Hierarchical predictive coding through dendritic error computation. Trends in Neurosciences 46(1):45–59

Miura SK, Scanziani M (2022) Distinguishing externally from saccade-induced motion in visual cortex. Nature 610(7930):135–142

Muzzu T, Saleem AB (2021) Feature selectivity can explain mismatch signals in mouse visual cortex. Cell Reports 37(1):109772

Polsky A, Mel BW, Schiller J (2004) Computational subunits in thin dendrites of pyramidal cells. Nat Neurosci 7(6):621–627

Pouget A, Dayan P, Zemel R (2000) Information processing with population codes. Nature Reviews Neuroscience 1(2):125–132

Tanabe S (2013) Population codes in the visual cortex. Neurosci Res 76(3):101–105

Vasilevskaya A, Widmer FC, Keller GB, et al (2023) Locomotion-induced gain of visual responses cannot explain visuomotor mismatch responses in layer 2/3 of primary visual cortex. Cell Reports 42(3):112096

Computational model of layer 2/3 in mouse primary visual cortex explains observed visuomotor mismatch response

Supplementary Information

Heiko Hoffmann[1*]

[1*]Magimine, LLC, P.O. Box 941154, Simi Valley, 93094, CA, USA.

Corresponding author(s). E-mail(s): heiko@magimine.com;

Abstract

This supplementary information shows three additional results: first, the mismatch responses for different velocity ranges encoded by the population code, second, the mismatch responses for a population code with bimodal tuning curves, and third, the mismatch responses when recording from neurons selected from three different population codes, each encoding the projection of a 2-dimensional velocity vector onto a basis vector that is different for each population code.

1 Dependence on velocity range

The main article used a fixed velocity-range offset of $v_{\text{Off}} = 0.76$. Here, I recreated the mismatch responses for different v_{Off} values (Fig. S1). For values near 0.76, the results changed only minimally compared to the main article. For $v_{\text{Off}} = -0.75$, the number of dMM neurons was largely reduced, only 3 out of 100, and their responses were not positively correlated with locomotion speeds anymore. For negative v_{Off}, the hMM neurons showed mismatch responses with larger variance across neurons.

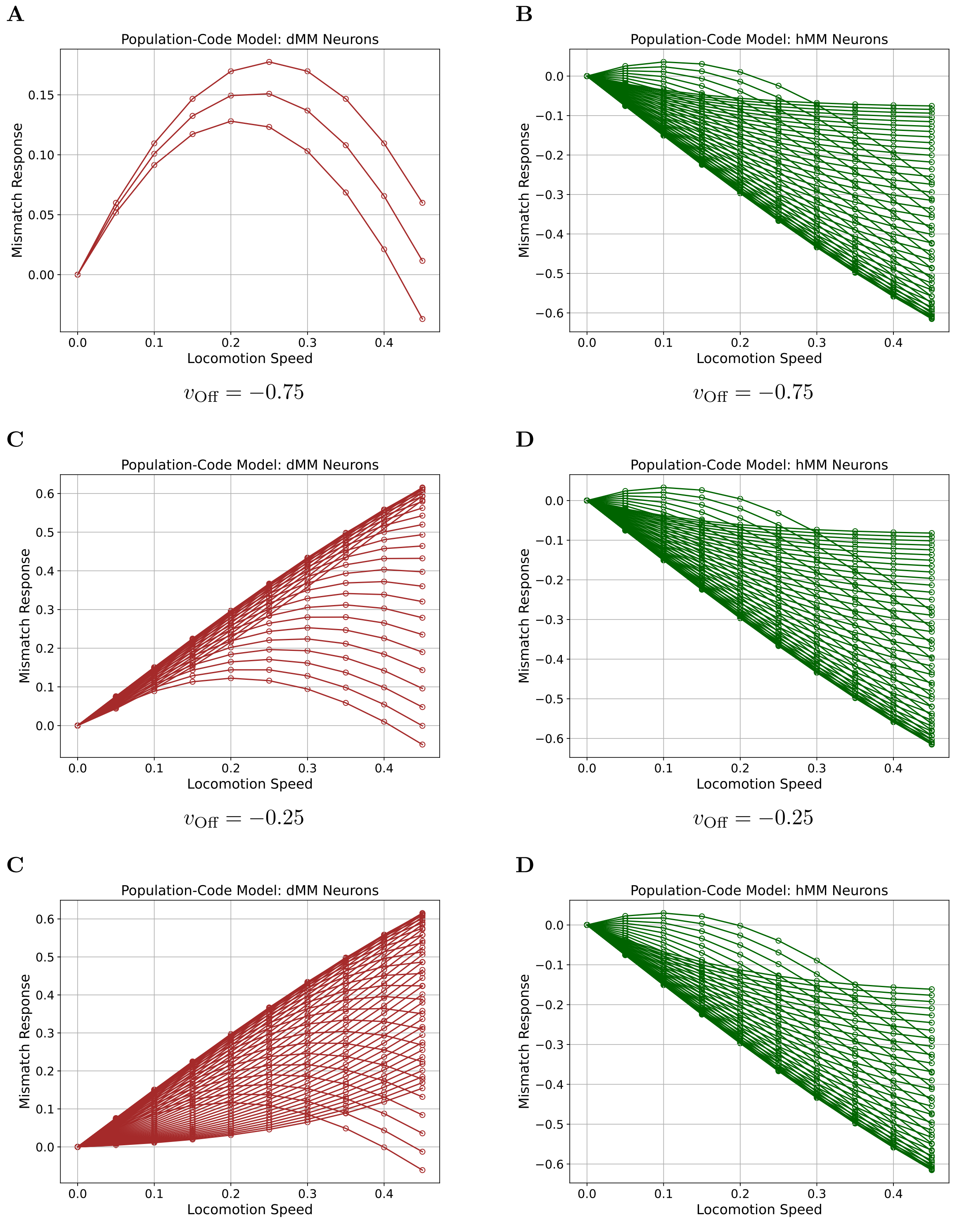
2 Bimodal tuning

I ran the simulation replacing the Gaussian tuning curves in our population-code model with cosine tuning curves,

$$A_i = \cos^2\left(2\pi(\Delta v - \mu_i)/3\right) \,, \tag{1}$$

where again μ_i is the preferred speed difference of neuron i. The values μ_i were uniformly distributed in the range from $-1 - v_{\text{Off}}$ to $1 - v_{\text{Off}}$ with $v_{\text{Off}} = 0.76$.

Figure S2 shows the results. The dMM and hMM neurons still displayed their main characteristic behavior: dMM mismatch response positively correlated with locomotion speed (linear fit with slope 1.34 ± 0.05) and hMM response negatively correlated (linear fit with slope -1.49 ± 0.04). Compared to Gaussian-tuning, the ratio of dMM and hMM neurons was different, and no value of the offset v_{Off} could be chosen to make the ratios match the mouse experiments. So, the population-code model with Gaussian tuning was in better agreement with experiment.



$v_{\text{Off}} = -0.75$ $v_{\text{Off}} = -0.75$

$v_{\text{Off}} = -0.25$ $v_{\text{Off}} = -0.25$

$v_{\text{Off}} = 0.25$ $v_{\text{Off}} = 0.25$

Fig. S1: Predicted scatterplot between locomotion speed, v_L, and mismatch response, ΔA, for different offset values, v_{Off}, of the population code. Each curve shows the response for a single neuron.

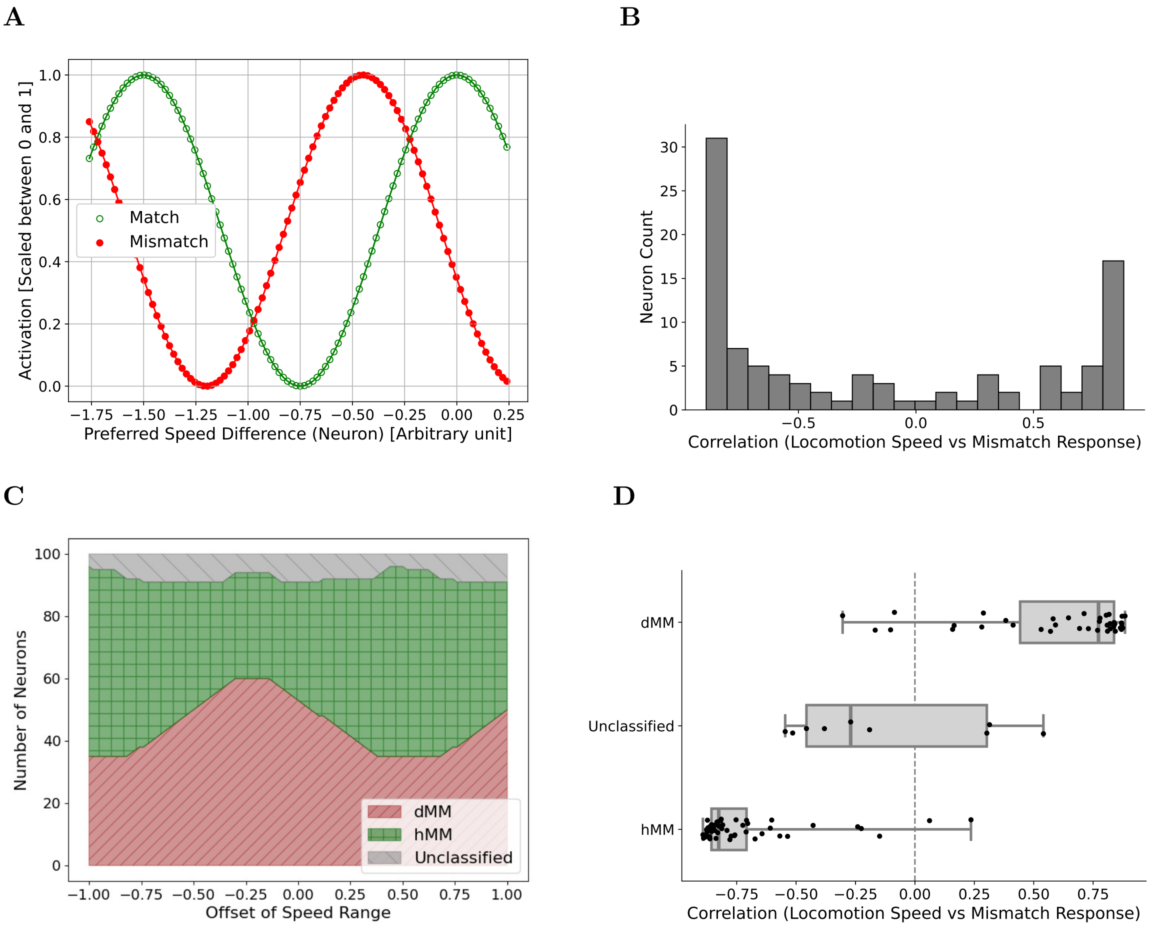


Fig. S2: Results for bimodal tuning. (A) The speed difference, Δv, was encoded by cosine tuning curves. (B) Correlation between mismatch response, ΔA, and locomotion speed, v_L. (C) The ratio of dMM and hMM neurons depends on the speed range, here, as a function of offset v_{Off}. (D) Boxplot of correlation depending on neural classification.

3 Multiple population codes

This section shows the results of a modified model using three instead of one population code. Each code encodes the velocity difference (prediction error) along a given direction. Here, I represented the velocity by a 2-dimensional vector,

$$\vec{v}_L = \begin{pmatrix} 0.05m \\ 0 \end{pmatrix} , \tag{2}$$

where m was an integer chosen from the interval 1 to 10, depending on the speed trial.

I used three basis vectors, as in Fig. S3, one for each population code. That is, we have an over-complete basis. I speculate that such encoding would be more likely because the redundancy increases robustness. To compute the value Δv_i represented by each population code, we project the locomotion velocity $\vec{v}_L$ onto the corresponding basis vector,

$$v_{L,i} = \vec{v}_L \cdot \vec{b_i} \, . \tag{3}$$

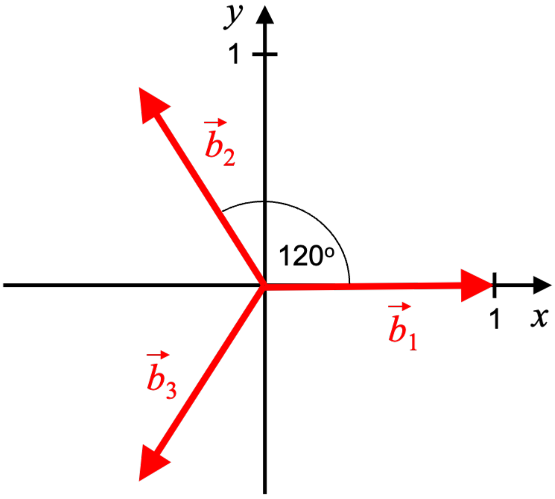


Fig. S3: Basis vectors for projecting the velocity vector onto 1-dimensional subspaces, one for each population code.

For each code, I used a velocity-range offset that was obtained by projecting the vector $(1.07, 0.6)$ onto the corresponding basis vector. This offset was chosen to yield dMM/hMM ratios as in the mouse experiments. I computed dMM and hMM neurons as in the main article, here, using a threshold of 0.02. This setting resulted in 157 dMM, 56 hMM, and 87 unclassified neurons.

Figure S4 shows the mismatch response results, which were similar to the ones presented in the main article. Different, here, we have three mismatch responses mixed together and each mismatch response had a different slope. The slope varied because the size of the projected speed values depended on the direction of the basis vector. Negative projected values, however, did not change the sign of the slope.

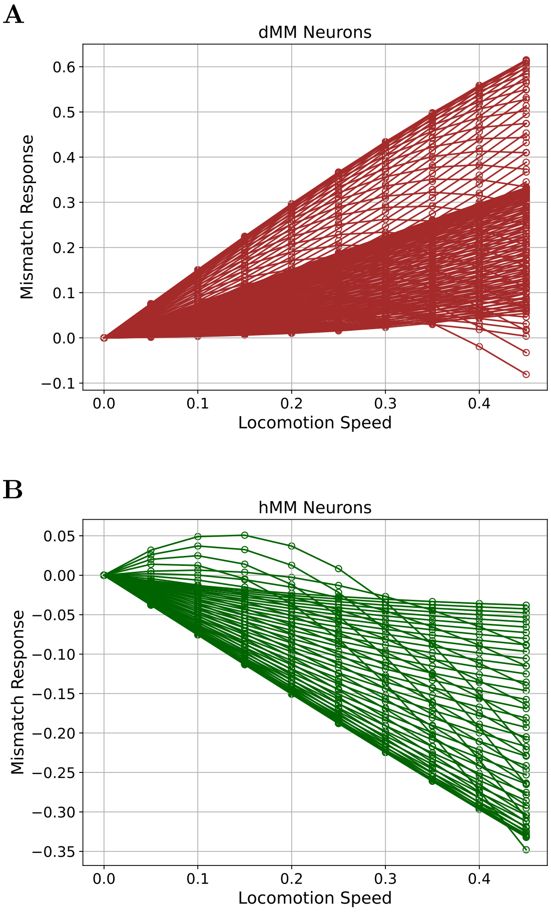


Fig. S4: Predicted scatterplot between locomotion speed and mismatch response for dMM (A) and hMM (B) neurons originating from three different population codes, each encoding the velocity difference along a given direction. Each curve shows the result from a single neuron.