

A neural data structure for novelty detection

Sanjoy Dasgupta^a, Timothy C. Sheehan^b, Charles F. Stevens^{c,d,1}, and Saket Navlakha^{e,1}

^aDepartment of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093; ^bNeurosciences Graduate Program, University of California, San Diego, La Jolla, CA 92093; ^cKavli Institute for Brain & Mind, University of California, San Diego, La Jolla, CA 92093; ^dMolecular Neurobiology Laboratory, The Salk Institute for Biological Studies, La Jolla, CA 92037; and ^eIntegrative Biology Laboratory, The Salk Institute for Biological Studies, La Jolla, CA 92037

Contributed by Charles F. Stevens, October 30, 2018 (sent for review August 22, 2018; reviewed by Piotr Indyk and Glenn Turner)

Novelty detection is a fundamental biological problem that organisms must solve to determine whether a given stimulus departs from those previously experienced. In computer science, this problem is solved efficiently using a data structure called a Bloom filter. We found that the fruit fly olfactory circuit evolved a variant of a Bloom filter to assess the novelty of odors. Compared with a traditional Bloom filter, the fly adjusts novelty responses based on two additional features: the similarity of an odor to previously experienced odors and the time elapsed since the odor was last experienced. We elaborate and validate a framework to predict novelty responses of fruit flies to given pairs of odors. We also translate insights from the fly circuit to develop a class of distance- and time-sensitive Bloom filters that outperform prior filters when evaluated on several biological and computational datasets. Overall, our work illuminates the algorithmic basis of an important neurobiological problem and offers strategies for novelty detection in computational systems.

fly olfactory circuit | computer science | data structures | Bloom filters | novelty detection

Assessing whether a stimulus is novel is a basic neural problem that helps alert organisms to new and potentially salient events (1, 2). This problem requires determining how much a given stimulus departs from those previously experienced. Here, we describe a simple data structure used by neural circuits to efficiently solve this problem.

The fruit fly determines the novelty of an odor using a two-step procedure. The first step involves assigning a “tag” (the term used in neurobiology) or “hash” (the term used in computer science) to the odor, where the tag/hash corresponds to a small set of neurons that fire action potentials in response to the odor (3–7). An odor is initially encoded in the fly’s nose as a 50-dimensional vector, corresponding to the firing rates of 50 different types of odorant receptor neurons (ORNs) (8). The 50 ORNs send odor information to 50 projection neuron types (PNs), whose responses are normalized such that the mean firing rate of the PNs is nearly the same for all odors and odor concentrations (6, 9). The PNs then project to about 2,000 Kenyon cells (KCs) that lie in a structure called the “mushroom body.” The connection matrix between the 50 PNs and the 2,000 KCs is sparse, approximately binary, and random (10). Each KC randomly selects about 6 of the 50 PNs and sums up their firing rates. Each KC then provides feedforward excitation to a single inhibitory neuron, which provides feedback inhibition to all of the KCs. As a result of this winner-take-all computation, only the top ~5% (100) of the highest-rate KCs fire in response to the odor; the remaining 95% are silenced. This 5% corresponds to the tag/hash of the odor. Thus, the hash of an odor is a sparse point in a 2,000-dimensional space. Each distinct odor activates a different 5% of the KCs, such that two very different odors will have little to no overlap in their active KCs.

The second step in determining odor novelty is to decide whether an odor, as defined by its hash, has been previously experienced. The output of the mushroom body is generated

by 34 mushroom body output neurons (MBONs) that receive input from the KCs and perform many different functions (11–13). Here, we are concerned with one such MBON, called MBON- $\alpha'3$ (14) [MBON- $\alpha'3$ actually consists of two distinct neurons that Hattori et al. (14) could not distinguish between; we follow their convention here and treat them together], which receives inputs from 350 of the 2,000 KCs (11) (see *SI Appendix* for more anatomical details). Based on a series of elegant experiments, Hattori et al. (14) showed that the activity of MBON- $\alpha'3$ in response to an odor encodes a novelty signal for the odor and that this signal is suppressed with odor familiarization (repeated presentations of the same odor).

Hattori et al. (14) proposed the following model of how MBON- $\alpha'3$ encodes novelty. For each odor presented to the fly, the hash for that odor is a sparse set of about 20 KCs (~5–10% of the 350 KCs that project to MBON- $\alpha'3$). Local dopamine released by a neuron called PPL1- $\alpha'3$ then modifies the strength of the 350 KC \rightarrow MBON- $\alpha'3$ synapses. Synapses made by the odor’s 20 activated KCs onto MBON- $\alpha'3$ weaken, whereas synapses made by nonactive KCs onto MBON- $\alpha'3$ strengthen. The activity of MBON- $\alpha'3$ is computed as the weighted sum of its inputs (i.e., the activity of each KC multiplied by its synaptic strength). Thus, repeated exposure to the same odor depresses active KC \rightarrow MBON- $\alpha'3$ synapses, which suppresses the activity of MBON- $\alpha'3$ in response to the odor, indicating that the odor has become familiar. At the same time, the synapses of nonactive KCs onto MBON- $\alpha'3$ strengthen, increasing over time the novelty response of odors with nonoverlapping hashes.

Significance

This work is significant for two reasons. First, considering fly olfaction in the context of a computational algorithm emphasizes features of olfactory processing that might otherwise be less apparent. For example, comparison of prior approaches for novelty detection makes clear that different odors are not just novel vs. familiar, but rather can have various degrees of novelty depending on the similarities between odors. Second, understanding how the fruit fly olfactory circuit detects novel odors can inspire new methods for similar problems in machine learning. In this way, both computer science and neuroscience benefit from the comparison of these two systems.

Author contributions: S.D., C.F.S., and S.N. designed research; S.D., T.C.S., and S.N. performed research; T.C.S. and S.N. analyzed data; and S.D., T.C.S., C.F.S., and S.N. wrote the paper.

Reviewers: P.I., Massachusetts Institute of Technology; and G.T., Janelia Research Campus.

The authors declare no conflict of interest.

Published under the [PNAS license](http://www.pnas.org/licenses).

¹To whom correspondence may be addressed. Email: navlakha@salk.edu or stevens@salk.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1814448115/-DCSupplemental.

Published online December 3, 2018.

Our Contributions

From a computer science perspective, we view the $KC \rightarrow MBON-\alpha'3$ synapses as storing a Bloom filter (Fig. 1). A Bloom filter is a space-efficient data structure that serves as the foundation for answering the question, “Does x belong to S ?”, where x is some query item (an odor) and S is a database of previously observed items. If x does not belong to S , then the item is considered novel, and vice versa. This question naturally comes up in many large-scale computing applications. For example, when Google indexes the web, its web crawler, upon reaching a website (x), needs to quickly determine whether it has visited this site in the past (S) to avoid the cost of reindexing. Similarly, determining whether a DNA sequence (x) belongs to a database (S) is a common primitive for many bioinformatics applications (15, 16).

The fly’s Bloom filter, however, modifies this basic question in three ways:

- i) Continuous-valued novelty. For the fly, novelty responses should reflect how different the query item is from those previously experienced. A traditional Bloom filter, on the other hand, reports only a binary response, indicating whether or not the exact item is in the database.
- ii) Distance sensitivity. The novelty of an odor with respect to a similar odor should be smaller than the novelty of the odor with respect to a very different odor. A traditional Bloom filter does not provide distance sensitivity, instead treating each pair of unique items as equidistant from all other pairs. Distance sensitivity also provides robustness, since in a noisy environment, it is unlikely that the fly will observe the exact same odor twice.
- iii) Time sensitivity. For the fly, if the same or a similar odor was last experienced a long time ago, its novelty should be larger than if the odor was experienced recently. A tradi-

tional Bloom filter, on the other hand, does not discount novelty over time.

The first two properties above modify the question to ask, “How close is x to some item in S ?” The third property asks, “How long ago was x , or some item similar to x , observed?” These modifications are also relevant practically. For example, when recommending news articles, one might ask, “How novel is this article compared with other articles the user has (recently) read?” or, “How different is this patient’s profile compared with others who have (recently) signed up for a clinical trial?” (15). While there are Bloom filters that satisfy properties two (15, 17–20) and three (21–23) separately, we are not aware of a Bloom filter that accounts for all three properties together. There are also technical differences between the fly’s Bloom filter and prior filters, which we elaborate below.

Here, we extend the framework of Hattori et al. (14) to quantitatively predict neural novelty responses based on the theory of Bloom filters. We also translate these three properties to extend a fundamental data structure to more application domains.

Results

The Mapping Between Bloom Filters and Novelty Detection Circuits. Here, we consider how to interpret the $KC \rightarrow MBON-\alpha'3$ synapses as a Bloom filter and how to modify traditional Bloom filters to take into account the three properties noted in the preceding section.

Consider the problem, “Does x belong to S ?” This problem could be easily solved by keeping a sorted list of all of the items in S and then performing a binary search for x or by inserting each item into a hash table and performing a lookup. The problem with these approaches is that they require storing all items in S in memory (e.g., in a list or a hash table), which can be prohibitively expensive for very large databases or if storing millions of small databases. Bloom filters are a space-efficient

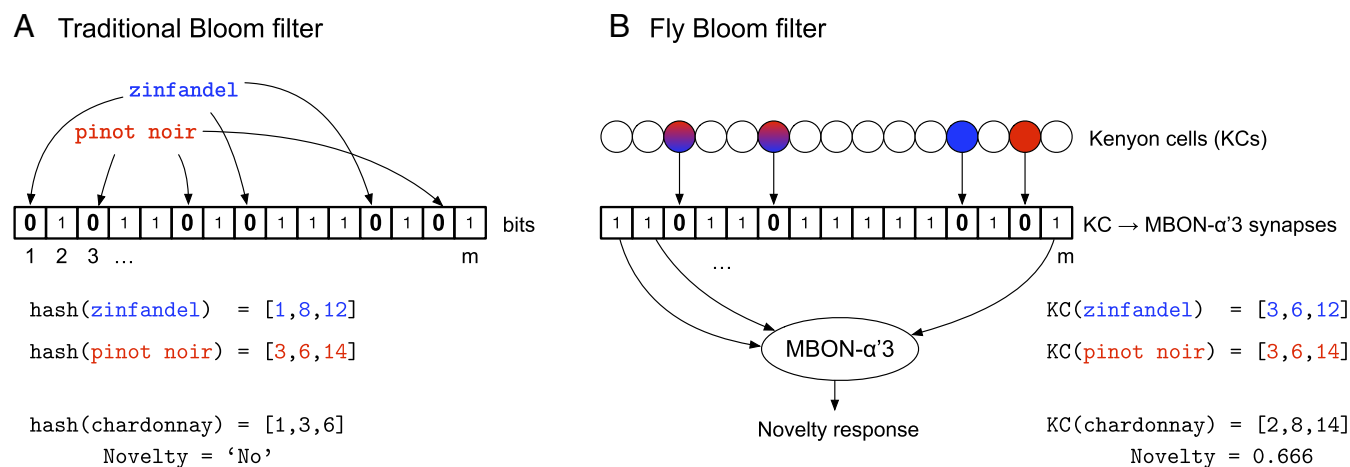


Fig. 1. Mapping between a traditional Bloom filter and the fly Bloom filter. (A) In a traditional Bloom filter, an array of m bits is used to store items. Initially, all bits are set to 1. To insert an item into the Bloom filter, the item is hashed k times using k independent hash functions. Each hash function maps the item to one bit in the array, and this bit is reset to 0. For example, the item “zinfandel” is hashed $k = 3$ times, which resets indexes 1, 8, and 12 in the filter. For “pinot noir,” indexes 3, 6, and 14 are reset. To test whether an item is novel, the item is hashed using the same k hash functions, and if all of the corresponding indexes are set to 0, the Bloom filter returns “No.” On the other hand, if any of its k hashed positions are 1, the Bloom filter returns “Yes.” This approach can lead to some false positives; e.g., “chardonnay” has not been inserted in the filter, but because each of its hash positions (1, 3, and 6) overlaps with positions reset for prior items, the Bloom filter incorrectly returns “No.” (B) In the fly Bloom filter, the weights of the m $KC \rightarrow MBON-\alpha'3$ synapses represent the m bits of the Bloom filter. Every odor activates k out of a population of m KCs. Each KC forms a synapse with the MBON- $\alpha'3$ neuron. When an odor is observed, the $KC \rightarrow MBON-\alpha'3$ synapses of the k active KCs are reset to 0. (With time sensitivity, these k synapses instead weaken, and the $m - k$ remaining synapses strengthen.) For example, the input zinfandel activates KCs 3, 6, and 12, causing the synapses from the 3rd, 6th, and 12th KC to MBON- $\alpha'3$ to reset to 0. Given an odor, MBON- $\alpha'3$ computes the weighted sum of its inputs as the novelty response for the odor. Because two of three active KCs for chardonnay (KCs 2 and 8) do not overlap with active KCs for previous odors, the novelty response for chardonnay is 0.666 (after normalization). The normalization step is used to restrict novelty values to the range [0, 1]. Because performance is measured as the correlation between the ground-truth novelty value (Eq. 1) and the predicted novelty value from the Bloom filter, the normalization does not affect model performance, nor is it necessarily required within the biological circuitry.

alternative that requires only a small number of bits per item to store, independent of the size or content of each individual item.

Formally, we are given a database of n items, $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ and a query item x . In a traditional Bloom filter, the goal is to output a binary novelty score, $f(x, \mathcal{S}) \in \{0, 1\}$ of x with respect to \mathcal{S} . The objective function (Does x belong to \mathcal{S} ?) is

$$f(x, \mathcal{S}) = \begin{cases} 0 & \text{if } x \in \mathcal{S} \\ 1 & \text{otherwise.} \end{cases}$$

For the fly, the goal is to output a continuous-valued novelty score, $f(x, \mathcal{S}) \in [0, 1]$ of x with respect to \mathcal{S} , and this score should be distance sensitive (we consider the third property, time sensitivity, later). Distance sensitivity means that f should reduce the novelty of x if there is an item similar to x in the database. The objective function (How close is x to some item in \mathcal{S} ?) is

$$f(x, \mathcal{S}) = \min_{s_i \in \mathcal{S}} d(x, s_i), \quad [1]$$

where $d(x, s_i)$ is a distance measure between items x and s_i . If $f(x, \mathcal{S}) = 1$, x is highly novel, whereas if $f(x, \mathcal{S}) = 0$, x is highly familiar.

The goal of a Bloom filter is to approximate $f(x, \mathcal{S})$ using a compressed m -bit representation of \mathcal{S} . In general, $m > n$, where n is the number of items in \mathcal{S} , and a typical value of m is $10n$ (i.e., 10 bits per item stored). In a traditional Bloom filter (Fig. 1A), each of the m bits is initialized to 1. (Conventionally, Bloom filters are initialized to 0 instead of to 1. We describe an “inverted” Bloom filter here so that novel items illicit a high novelty response instead of a low response, but the argument remains the same.) Each $s \in \mathcal{S}$ is inserted into the filter by applying $k \ll m$ independent hash functions to s . Each hash function maps s to one of the m positions in the Bloom filter, and each of these positions is reset to 0. To determine the novelty of a query x , one applies the same k hash functions to x to obtain k positions in the Bloom filter; if any of these bits are set to 1, then the item definitely does not belong to \mathcal{S} . If all of the bits are set to 0, then the item probably belongs to \mathcal{S} . This procedure can induce some false positives if, for example, the bits probed for x collectively overlap with the bits reset for items previously inserted (e.g., see Fig. 1A for chardonnay).

For the fly, we propose that the KC \rightarrow MBON- $\alpha'3$ synapses store a Bloom filter for novelty detection (Fig. 1B). Here, the m KC \rightarrow MBON- $\alpha'3$ synapses represent the m bits of storage in the Bloom filter. The indexes of the k active KCs for an odor act as the k hashed positions of the odor. When an odor is presented (i.e., inserted into the filter), the strength of each of the m synapses is modified. Let $w(i)$ correspond to the weight of the i^{th} KC \rightarrow MBON- $\alpha'3$ synapse. Initially, each $w(i) = 1$. After an odor x is observed, each synaptic weight changes as follows:

$$w(i) = \begin{cases} w(i) \times \delta & \text{if the } i^{\text{th}} \text{ KC is active for } x \\ w(i) + \epsilon & \text{if the } i^{\text{th}} \text{ KC is not active for } x. \end{cases} \quad [2]$$

In other words, the synaptic weights of the k active KCs for the odor each decay by some factor, $0 \leq \delta < 1$. The remaining $m - k$ KCs strengthen by a small amount, $0 \leq \epsilon \leq 1$. [Here, we assume that each KC’s activity for an odor is binary (on or off), which is an approximation (SI Appendix).] When ignoring time, we set $\delta = 0$ and $\epsilon = 0$ so that once an odor activates a KC, its weight gets reset to 0, and no recovery occurs with the presentation of subsequent odors. Thus, all synaptic weights are binary (bits), as in a traditional Bloom filter. See SI Appendix, Fig. S3 for justification of these functional forms for recovery and decay.

To generate a continuous-valued novelty score of an odor x from the Bloom filter, we sum the weights of the k active KC \rightarrow MBON- $\alpha'3$ synapses for odor x and normalize by k to generate

novelty responses between 0 and 1 (Fig. 1B). Because all weights are binary (ignoring time), the only weights that will be 1 are those corresponding to KCs that are active for x and no other odor in \mathcal{S} previously inserted into the filter.

Predicting Neural Novelty Responses to Odor Pairs. Using the reasoning above, the novelty response of an odor with respect to a single previous odor should be proportional to the percentage of KCs shared by the two odors: higher overlap \rightarrow smaller novelty response. We tested this hypothesis using neural recordings of MBON- $\alpha'3$ in response to pairs of odors, where one odor was presented, and then the activity of MBON- $\alpha'3$ was recorded in response to a second odor (14) (SI Appendix, Table S1). In this setting, $|\mathcal{S}| = 1$ (containing the first odor), and we wish to predict the novelty of x , a second odor. Hattori et al. (14) did not describe or analyze how odor similarity might affect novelty responses in the fly.

To determine the similarity of two odors below, we used a publicly available database [DoOR 2.0 (24)], which provides ORN responses to different odors by collecting experimental data from many studies.

First, we tested how odor similarity, defined based on the differences in ORN activity for the two odors, affected novelty responses. Specifically, each odor was associated with a vector of recorded ORN activity in response to the odor, using the DoOR database (24). We found a significant correlation between the Euclidean distance between the vectors and the recorded novelty response of MBON- $\alpha'3$ for the pair: Higher distance \rightarrow larger novelty ($r = 0.91$; Fig. 2A).

Second, we tested how well the predicted novelty score from the fly Bloom filter, defined based on the overlap of active Kenyon cells for the pair, correlated with the recorded novelty response. Specifically, the KC responses for each odor were computed by multiplying the vector of ORN activity for the odor by a sparse, binary random projection matrix (7, 10) into an $m = 350$ -dimensional space. We then set the indexes of the $k = 20$ ($\approx 5\%$ of 350) most rapidly firing KCs to 1 and set the rest of the indexes to 0. We again found that the more overlap in active KCs shared by the pair, the lower the novelty response for the pair ($r = -0.94$; Fig. 2B). Due to the small number of odor pairs available, we tested how robust these correlations were, using leave-one-out cross-validation, and found no change to our conclusions (SI Appendix, Table S3).

Overall, these results show that the similarity between odors—calculated either in input ORN space or in KC hash space—could predict the novelty response for the pair. Further, the similarity between two odors can be read off from the output of the Bloom filter in a manner that is consistent with the first and second properties that we proposed.

Analytically Predicting Novelty Responses. In SI Appendix, we mathematically derive the expected novelty response from the fly Bloom filter for an item x with respect to a set of previously observed items \mathcal{S} . We also instantiate bounds on the expected novelty responses for the cases where inputs are binary or exponentially distributed [as in faces (25) and odors (6)] (SI Appendix, Fig. S1).

Empirical Evaluation on Benchmark Datasets. To translate these neural insights to the computational domain, we compared three types of Bloom filters—the fly Bloom filter, a traditional Bloom filter, and a prior distance-sensitive Bloom filter (LSBF) (15, 17) (SI Appendix, Algorithms S1–S3). We compared performance on five datasets (SI Appendix, Table S2); for each dataset, we used 10-fold cross-validation, where all items from the training set \mathcal{S} are inserted into the Bloom filter. Then, for each item in the test set, we computed its ground-truth novelty response via Eq. 1 and the predicted novelty from the filter, and we report

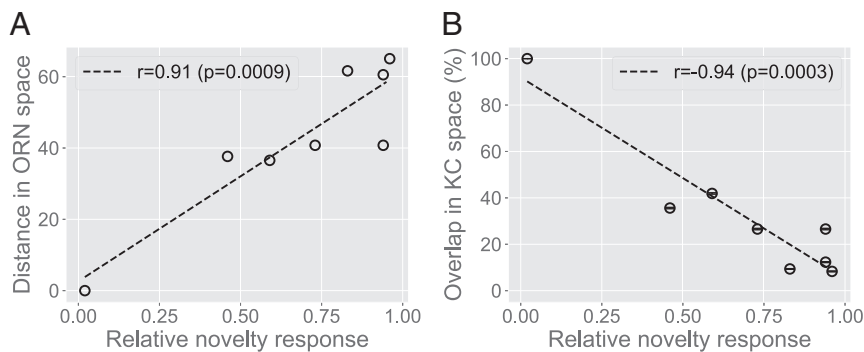


Fig. 2. Predicting neural novelty responses. In *A* and *B*, open circles represent a pair of odors. (*A*) The x axis is the recorded MBON- α' 3 novelty response of a second odor after repeated presentations of a first odor (14). The reported value is normalized to the maximum response over the trial. This “relative novelty response” indicates the novelty of a second odor with respect to that of the first odor, averaged over 10 trials. The y axis is the Euclidean distance between the vectors of ORN firing activity for the two odors. The larger the Euclidean distance is (i.e., the more dissimilar the odors), the larger the novelty response. (*B*) The x axis is the MBON- α' 3 novelty response, and the y axis is the percentage of KCs that are active for both odors. The more overlap between KCs (i.e., the more similar the odors), the smaller is the novelty response. Error bars denote SEM over 20 random projection matrices used to compute KC activities; errors appear smaller than the plotting symbol. In both *A* and *B*, the dashed line shows a least-squares regression of the points, with correlation coefficient and *P* value at the top.

the correlation between the ground-truth and predicted novelty responses over all items in the test set. We set $m = 30n$ and varied $k \in [5, 50]$.

First, we evaluated performance on two neural datasets (Fig. 3 *A* and *B*). The first one contains neural activity of 24 ORNs in the fruit fly in response to $n = 110$ odors (8). The second one contains neural activity of 99 face patch neurons in the macaque in response to $n = 2,000$ faces (26). In both cases, we found sig-

nificant improvement in novelty detection using the fly Bloom filter compared with the alternatives. For example, on the odors dataset, the correlation between ground-truth and predicted novelty using the fly Bloom filter was 0.657 ± 0.06 compared with 0.537 ± 0.08 using the LSBF ($k = 40$). Similarly, on the face data, the correlations were 0.508 ± 0.03 for the fly vs. 0.404 ± 0.03 for the LSBF. On all datasets, the traditional Bloom filter had close to zero correlation between ground-truth and predicted novelty

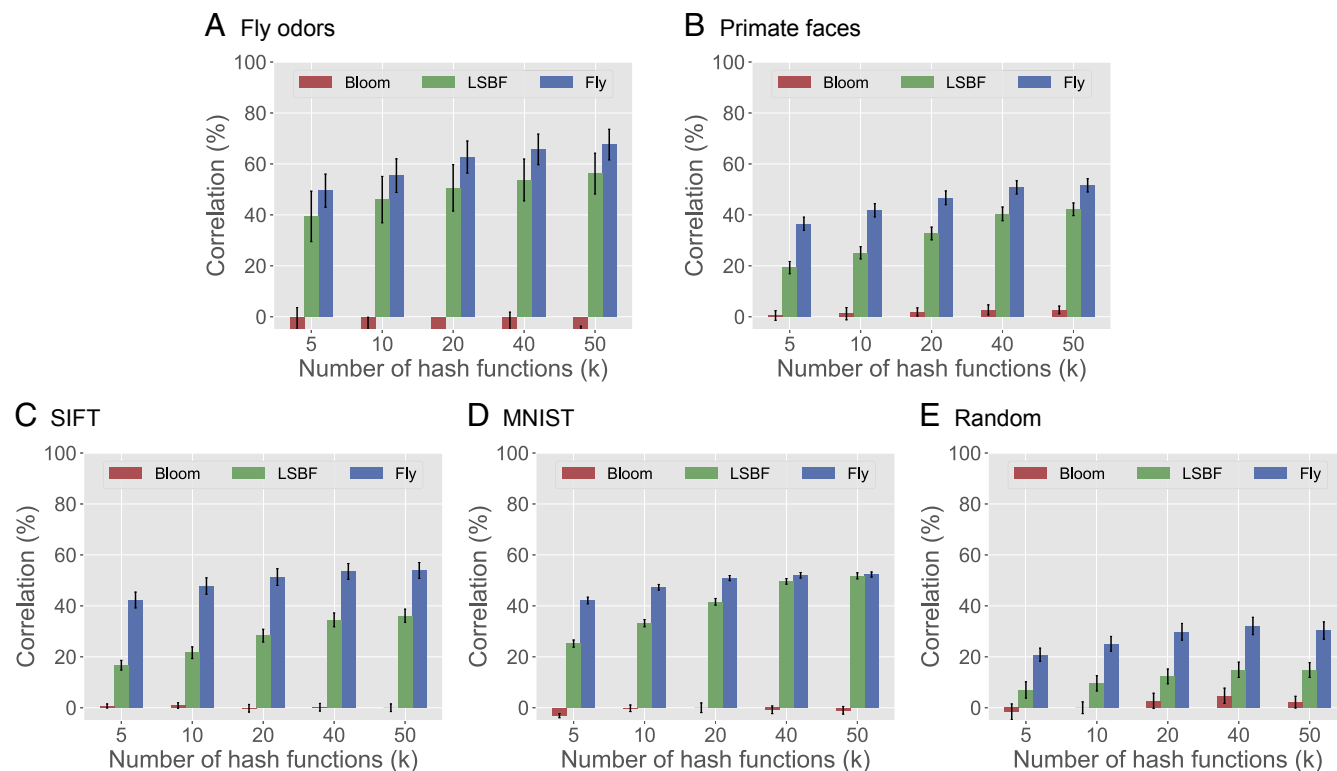


Fig. 3. Empirical comparison of novelty detection for three Bloom filters on five datasets. In *A–E*, the x axis shows the number of hash functions k used in the Bloom filter, and the y axis shows the correlation (higher is better) between the ground-truth novelty response and the predicted novelty response by the Bloom filter. Three filters are compared: Bloom (a traditional Bloom filter), LSBF [a prior distance-sensitive Bloom filter (15, 17)], and Fly (our proposed fly Bloom filter). The datasets are as follows: (*A*) Fruit fly odors, $n = 110$ odors, $d = 24$ ORN responses; (*B*) primate faces, $n = 2,000$ faces, $d = 99$ face patch neuron responses; (*C*) SIFT images, $n = 5,000$ images, $d = 128$ features; (*D*) MNIST images, $n = 5,000$ images, $d = 784$ features; and (*E*) random exponentially distributed data, $n = 1,000$ vectors, $d = 20$ dimensions. On all datasets, the fly Bloom filter outperforms, or performs no worse than, the other filters.

because the hash functions used in a traditional Bloom filter are random and not distance sensitive.

Second, we tested performance on three computational datasets. The first two are image datasets [scale-invariant feature transform descriptors (SIFT) and modified National Institute of Standards and Technology database (MNIST)] with $n = 5,000$ each, and the third one is random data drawn from an exponential distribution (Fig. 3 C–E). In all three cases, the fly Bloom filter outperforms, or performs no worse than, other filters. For example, on the SIFT dataset, the fly Bloom filter achieved a correlation of 0.535 ± 0.03 vs. 0.345 ± 0.03 and 0.002 ± 0.02 for the LSBF and traditional filters, respectively. We also repeated all experiments with $m = 10n$ and found similar results (SI Appendix, Fig. S2).

Overall, the fly Bloom filter enhances novelty detection vs. that of other filters, while using the same space-efficient representation.

Building Distance- and Time-Sensitive Bloom Filters. Similar items observed recently in time should illicit a lower novelty response than similar items observed far apart in time. This means that old data stored in the Bloom filter should slowly decay or be evicted from the filter over time. Eviction is also important in life-long learning applications, where the database is not of fixed size but continuously grows (21–23); unless old data are forgotten, the Bloom filter would eventually fill up such that every bit is reset to zero.

Currently, there is little biological knowledge of how distance sensitivity and time sensitivity combine to create a single novelty response in the fruit fly (Discussion). In SI Appendix, we propose candidate objective functions and show that the fly Bloom filter also works well in this scenario (SI Appendix, Table S4).

Discussion

The fruit fly brain evolved a simple space-efficient data structure to compute the novelty of odors, directly taking advantage of sparse, high-dimensional representations (27–29) useful for discriminating odors (7, 30). Our framework can be used to predict novelty responses in fruit flies to odor pairs (14) and can be used to enhance novelty detection for computational applications (31, 32), including for anomaly detection (33), computer networking (34), and deep learning (35, 36).

There are many variants of Bloom filters that extend the basic functionality of traditional Bloom filters. These include filters that adapt k , the number of hash functions used, based on the popularity of the data (37, 38), that decentralize the filter to handle failures (2, 39), that allow items to be deleted (16, 40, 41), and that use other sophisticated inference steps (42). In practice, some of these features may need to be ported to the fly's Bloom filter. There are also several methods used to compress high-dimensional datasets that enable estimating $f(x, S)$ efficiently using a nearest-neighbors search (43). For example, Jégou et al. (44) learn short codes for dataset items using product quantization techniques, and Johnson et al. (45) further speed up these computations using GPUs. Indyk and Wagner (46) develop a space-efficient tree-based data structure for storing items that requires less bits per item to store than prior structures. Compared with the approach presented here, these methods are not based on Bloom filters, are not dynamic (time sensitive), and do not appear to be biologically plausible. Nonetheless, in practical applications of novelty detection, these approaches should also be considered.

Our work raises three questions about timing mechanisms and their effect on novelty calculations:

- i) The rates of synapse weight recovery (ϵ) and decay (δ) may provide clues of how the brain recovers and suppresses novelty over time. Based on experimental data, the functional forms of these two parameters appear to follow an additive-increase, multiplicative-decrease mechanism (SI Appendix, Fig. S3); interestingly, multiplicative weight update rules are also commonly used by online learning algorithms to converge to optimal decisions over time (47–49). While this is one possible mathematical form for recovery and decay, for the fly Bloom filter to compute novelty values as one might expect, two general rules seem to apply. First, decay after initial exposure to an odor should be aggressive. If decay were slow, then the novelty signal would persist over several successive presentations of an odor, which may unnecessarily burden attention. Further, aggressive decay immediately heightens the difference between an odor observed once recently (familiar) vs. an odor that has not been observed for a long time (novel). Second, recovery from familiarity back to novelty should be relatively slower. Otherwise, very soon after observing an odor, the odor would become novel again, limiting the timescale over which novelty can be integrated. More work, however, is needed to test these rules behaviorally.
- ii) More information is needed about how novelty calculations trade off distance vs. time; i.e., how can the fruit fly distinguish between two similar odors observed far apart in time vs. two dissimilar odors observed recently in —both of which may illicit a high novelty response? Might other MBONs be involved? Computationally, given the local synaptic weight update rule (Eq. 2), can we reverse engineer the global objective function that the local rule is trying to optimize?
- iii) The amount of dopamine released by PPL1- $\alpha'3$ for an odor, which may affect recovery and decay rates, could be related to the odor's concentration or valence and may be regulated by feedback from other MBONs (50); at the extreme, if no dopamine is released for an odor, then this might imply that the odor was not “inserted into the filter.”

Understanding these parameters may allow novelty responses to be predicted for a time sequence of odors, instead of simply odor pairs.

The theory of Bloom filters presented here may also help inform studies of novelty detection in other species and brain regions. Evidence points to similar patterns of decay and recovery of novelty responses, albeit over longer timescales, in the vertebrate olfactory system (51). Novelty detection also occurs in the hippocampus (52), in the amygdala (53), and in auditory systems (54), although circuit mechanisms here are less known.

Materials and Methods

In SI Appendix, we describe (i) additional background on the fly olfactory and novelty detection circuits, (ii) the data used for analyzing novelty responses to odor pairs and cross-validation results, (iii) the three Bloom filters used in our empirical comparison, (iv) computational analysis of distance- and time-sensitive Bloom filters, and (v) theoretical analysis of the fly Bloom filter for predicting novelty responses.

ACKNOWLEDGMENTS. The authors thank John Berkowitz, Shyam Srinivasan, and Jaiyam Sharma for helpful comments on the manuscript. S.N. was supported by the Pew Charitable Trusts, the Kavli Institute for Brain and Mind, and the National Institute on Deafness and Other Communication Disorders (NIDCD) of the National Institutes of Health under Award 1R01DC017695.

1. Ranganath C, Rainer G (2003) Neural mechanisms for detecting and remembering novel events. *Nat Rev Neurosci* 4:193–202.
2. Koloniari G, Ntarmos N, Pitoura E, Souravlias D (2011) One is enough: Distributed filtering for duplicate elimination. *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM '11*. Available at <https://dl.acm.org/citation.cfm?id=2063643>. Accessed April 2, 2018.

3. Leonardo A (2005) Degenerate coding in neural systems. *J Comp Physiol A Neuroethol Sens Neural Behav Physiol* 191:995–1010.
4. Campbell RA, et al. (2013) Imaging a population code for odor identity in the *Drosophila* mushroom body. *J Neurosci* 33:10568–10581.
5. Turner GC, Bazhenov M, Laurent G (2008) Olfactory representations by *Drosophila* mushroom body neurons. *J Neurophysiol* 99:734–746.

6. Stevens CF (2015) What the fly's nose tells the fly's brain. *Proc Natl Acad Sci USA* 112:9460–9465.
7. Dasgupta S, Stevens CF, Navlakha S (2017) A neural algorithm for a fundamental computing problem. *Science* 358:793–796.
8. Hallem EA, Carlson JR (2006) Coding of odors by a receptor repertoire. *Cell* 125:143–160.
9. Olsen SR, Bhandawat V, Wilson RI (2010) Divisive normalization in olfactory population codes. *Neuron* 66:287–299.
10. Caron SJ, Ruta V, Abbott LF, Axel R (2013) Random convergence of olfactory inputs in the *Drosophila* mushroom body. *Nature* 497:113–117.
11. Aso Y, et al. (2014) The neuronal architecture of the mushroom body provides a logic for associative learning. *Elife* 3:e04577.
12. Hige T, Aso Y, Modi MN, Rubin GM, Turner GC (2015) Heterosynaptic plasticity underlies aversive olfactory learning in *Drosophila*. *Neuron* 88:985–998.
13. Takemura SY, et al. (2017) A connectome of a learning and memory center in the adult *Drosophila* brain. *Elife* 6:26975.
14. Hattori D, et al. (2017) Representations of novelty and familiarity in a mushroom body compartment. *Cell* 169:956–969.
15. Kirsch A, Mitzenmacher M (2006) Distance-sensitive Bloom filters. *Proceedings of the Meeting on Algorithm Engineering & Experiments, ALENEX '06*. Available at <https://epubs.siam.org/doi/abs/10.1137/1.9781611972863.4>. Accessed April 11, 2018.
16. Pandey P, Bender MA, Johnson R, Patro R (2017) A general-purpose counting filter: Making every bit count. *Proceedings of the ACM International Conference on Management of Data, SIGMOD '17*. Available at <https://dl.acm.org/citation.cfm?id=3035963>. Accessed April 19, 2018.
17. Hua Y, Xiao B, Veeravalli B, Feng D (2012) Locality-sensitive Bloom filter for approximate membership query. *IEEE Trans Comput* 61:817–830.
18. Goswami M, Pagh R, Silvestri F, Sivertsen J (2017) Distance sensitive Bloom filters without false negatives. *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '17*. Available at <https://dl.acm.org/citation.cfm?id=3039703>. Accessed May 2, 2018.
19. Manber U, Wu S (1994) An algorithm for approximate membership checking with application to password security. *Inf Process Lett* 50:191–197.
20. Qian J, Zhu Q, Chen H (2015) Multi-granularity locality-sensitive Bloom filter. *IEEE Trans Comput* 64:3500–3514.
21. Deng F, Rafiei D (2006) Approximately detecting duplicates for streaming data using stable Bloom filters. *Proceedings of the ACM International Conference on Management of Data, SIGMOD '06*. Available at <https://dl.acm.org/citation.cfm?id=1142477>. Accessed April 24, 2018.
22. Zhang L, Guan Y (2008) Detecting click fraud in pay-per-click streams of online advertising networks. *The 28th International Conference on Distributed Computing Systems, ICDCS '08*. Available at <https://ieeexplore.ieee.org/document/4595871>. Accessed April 12, 2018.
23. Zhao Y, Wu J (2010) B-sub: A practical Bloom-filter-based publish-subscribe system for human networks. *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems, ICDCS '10*. Available at <https://ieeexplore.ieee.org/document/5541685>. Accessed May 30, 2018.
24. Munch D, Galizia CG (2016) DoOR 2.0—comprehensive mapping of *Drosophila melanogaster* odorant responses. *Sci Rep* 6:21841.
25. Stevens CF (2018) Conserved features of the primate face code. *Proc Natl Acad Sci USA* 115:584–588.
26. Chang L, Tsao DY (2017) The code for facial identity in the primate brain. *Cell* 169:1013–1028.
27. Rinkus GJ (2014) Sparsify: Event recognition via deep hierarchical sparse distributed codes. *Front Comput Neurosci* 8:160.
28. Babadi B, Sompolinsky H (2014) Sparseness and expansion in sensory representations. *Neuron* 83:1213–1226.
29. Cui Y, Ahmad S, Hawkins J (2017) The HTM spatial pooler—A neocortical algorithm for online sparse distributed coding. *Front Comput Neurosci* 11:111.
30. Lin AC, Bygrave AM, de Calignon A, Lee T, Miesenbock G (2014) Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination. *Nat Neurosci* 17:559–568.
31. Pimentel MAF, Clifton DA, Clifton L, Tarassenko L (2014) Review: A review of novelty detection. *Signal Process* 99:215–249.
32. Markou M, Singh S (2003) Novelty detection: A review—Part 2: Neural network based approaches. *Signal Process* 83:2499–2521.
33. Luo C, Shrivastava A (2018) Arrays of (locality-sensitive) count estimators (ACE): Anomaly detection on the edge. *Proceedings of the 2018 World Wide Web Conference, WWW '18*. Available at <https://dl.acm.org/citation.cfm?id=3186056>. Accessed June 8, 2018.
34. Broder A, Mitzenmacher M (2004) Network applications of Bloom filters: A survey. *Internet Math* 1:485–509.
35. Serrà J, Karatzoglou A (2017) Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks. arXiv:1706.0399. Preprint, posted June 13, 2017.
36. Salvi A, Ercoli S, Bertini M, Bimbo AD (2016) Bloom filters and compact hash codes for efficient and distributed image retrieval. arXiv:1605.00957. Preprint, posted May 3, 2016.
37. Bruck J, Gao J, Jiang A (2006) Weighted Bloom filter. *IEEE International Symposium on Information Theory*. Available at <https://ieeexplore.ieee.org/document/4036381>. Accessed April 22, 2018.
38. Zhong M, Lu P, Shen K, Seiferas J (2008) Optimizing data popularity conscious Bloom filters. *Proceedings of the 27th ACM Symposium on Principles of Distributed Computing, PODC '08*. Available at <https://dl.acm.org/citation.cfm?id=1400798>. Accessed May 30, 2018.
39. Laufer RP, Velloso PB, Duarte OCMB (2011) A generalized Bloom filter to secure distributed network applications. *Comput Netw* 55:1804–1819.
40. Fan L, Cao P, Almeida J, Broder AZ (2000) Summary cache: A scalable wide-area web cache sharing protocol. *IEEE/ACM Trans Netw* 8:281–293.
41. Fan B, Andersen DG, Kaminsky M, Mitzenmacher MD (2014) Cuckoo filter: Practically better than Bloom. *Proceedings of the 10th ACM International Conference on Emerging Networking Experiments and Technologies, CoNEXT '14*. Available at <https://dl.acm.org/citation.cfm?id=2674005.2674994>. Accessed April 2, 2018.
42. Dautrich JL Jr, Ravishanker CV (2013) Inferential time-decaying Bloom filters. *Proceedings of the 16th International Conference on Extending Database Technology, EDBT '13*. Available at <https://dl.acm.org/citation.cfm?id=2452376.2452405>. Accessed June 4, 2018.
43. Wang J, Liu W, Kumar S, Chang S (2016) Learning to hash for indexing big data—a survey. *Proc IEEE* 104:34–57.
44. Jegou H, Douze M, Schmid C (2011) Product quantization for nearest neighbor search. *IEEE Trans Pattern Anal Mach Intell* 33:117–128.
45. Johnson J, Douze M, Jégou H (2017) Billion-scale similarity search with GPUs. arXiv:1702.08734. Preprint, posted February 28, 2017.
46. Indyk P, Wagner T (2018) Approximate nearest neighbors in limited space. *Proceedings of the 31st Conference On Learning Theory (COLT), Proceedings of Machine Learning Research*, eds Dy J, Krause A (Proceedings of Machine Learning Research, Stockholm), Vol 75, pp 2012–2036.
47. Arora S, Hazan E, Kale S (2012) The multiplicative weights update method: A meta-algorithm and applications. *Theor Comput* 8:121–164.
48. Suen JY, Navlakha S (2017) Using inspiration from synaptic plasticity rules to optimize traffic flow in distributed engineered networks. *Neural Comput* 29:1204–1228.
49. Chiu DM, Jain R (1989) Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Comput Netw ISDN Syst* 17:1–14.
50. Eichler K, et al. (2017) The complete connectome of a learning and memory centre in an insect brain. *Nature* 548:175–182.
51. Kato HK, Chu MW, Isaacson JS, Komiyama T (2012) Dynamic sensory representations in the olfactory bulb: Modulation by wakefulness and experience. *Neuron* 76:962–975.
52. Knight R (1996) Contribution of human hippocampal region to novelty detection. *Nature* 383:256–259.
53. Blackford JU, Buckholtz JW, Avery SN, Zald DH (2010) A unique role for the human amygdala in novelty detection. *Neuroimage* 50:1188–1193.
54. Malmierca MS, Sanchez-Vives MV, Escera C, Bendixen A (2014) Neuronal adaptation, novelty detection and regularity encoding in audition. *Front Syst Neurosci* 8:111.