THE UNIVERSITY OF CHICAGO

RELIABLE AND CONTEXT-DEPENDENT COMPUTATION IN THE BRAIN

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE BIOLOGICAL SCIENCES
AND THE PRITZKER SCHOOL OF MEDICINE
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

COMMITTEE ON COMPUTATIONAL NEUROSCIENCE

BY
WILLIAM JEFFREY JOHNSTON

CHICAGO, ILLINOIS
DECEMBER 2020

"To think without the world, is that possible? Is thinking more material than we know? There are affinities that escape our perception: the unknown is an immense reality."

– Etel Adnan, *Fog*

Table of Contents

List of Figures

ACKNOWLEDGMENTS

I would like to thank my adviser David Freedman for the advice, encouragement, and training that he provided. In one way or another, he was instrumental in the completion of every aspect of this work. I would also like to thank Stephanie Palmer for the additional training and advice that she provided, both as part of her service on my committee and as a close collaborator and secondary mentor. I would also like to thank the other members of my committee, John Maunsell and Ed Awh, both of whom have given guidance and advice that have strengthened different parts of this and other work.

Additionally, I would like to thank Krithika Mohan, Barbara Peysakhovich, Stephanie Tetrick, Yang Zhou, Nicolas Masse, and Eric Potash for additional scientific guidance and technical assistance.

Finally, I would like to thank my friends and family for their support and understanding.

ABSTRACT

The brain is a distributed computational system. While the brain has been understood to exhibit at least weak modularity for over a century, numerous important questions about the degree and consequences of that modularity remain. My thesis work consisted of three projects, which are each related to distinct questions about the nature and function of modularity in the brain.

First, I investigated how the neural code within distinct regions of the brain could be made reliable in the face of the unreliability of individual neurons. In this work, I found that increasing the dimensionality of neural representations through conjunctive mixing of multiple stimulus features improves the reliability of those representations by orders of magnitude relative to representations without mixing. This work provides an explanation for a commonly observed phenomenon in experiments: The apparent random conjunctive mixing of stimulus features in single neurons. However, it also intersects with questions about modularity. In particular, the benefits that can be derived from this conjunctive mixing depend strongly on the size of the neural population available to participate in the representation – that is, on the size of a particular brain region. Further work will explore how this pressure for larger regions is tempered by other constraints in the brain.

Second, I investigated how the neural code across distinct regions of the brain could be made reliable in the presence of multiple heterogeneous objects that are represented only partially within each region. As an example, two cats in the world have both visual and auditory representations in the brain. To guide behavior, the brain must integrate these different facets of the same animals. We describe the necessary conditions for this integration process to be reliable. Further, we outline a tradeoff, in which the fidelity of each individual representation can be increased at the cost of a greater risk of catastrophic integration errors, in which the auditory features of one cat are integrated with the visual features of the other

cat. More generally, this work provides another constraint on the modularity exhibited by the brain. We show that redundancy in the information represented by distinct brain regions is absolutely necessary for reliable integration. Thus, this work illustrates a pressure for the brain to use fewer distinct modules, so that it can satisfy the overall goal of redundancy reduction for producing efficient neural codes.

Third, I performed electrophysiological experiments to investigate the role of a particular brain region, the lateral intraparietal area (LIP), in two distinct tasks. These experiments revealed that the putative function of LIP in the representation of both visually guided actions and the behavioral relevance of different parts of the visual field is highly task-dependent. In particular, our results indicate that LIP may serve this role primarily in the context of directed tasks, while it is less engaged in undirected, free-viewing behavior. These results illustrate the extreme context-dependence of many of our inferences about the functional role of different brain regions.

Together, my thesis work refines our understanding of the role and reason for modularity in the brain – and points to numerous directions for future work. In particular, my work provides many of the necessary tools for beginning to build a more comprehensive normative theory of neural modularity that will be necessary to a comprehensive understanding of the brain.

CHAPTER 1

INTRODUCTION

1.1 Distributed computation and modularity in the brain

The structure of the brain has long been understood to dictate much about its function. In some cases, the relationship of structure to function is relatively direct: For instance, an important cue for horizontal sound localization is the difference in onset time of the sound to each ear. If the sound arrives earlier in the left ear, then the sound likely originates from a point in space to the left of the head – and the precise difference in onset can be used to localize the origin point with high accuracy on the horizontal plane. In the barn owl, auditory localization is performed through neural "delay lines" that are exquisitely designed to exploit the information contained in this onset difference. In particular, neural transmission delays and physical distance in the brain are both designed so that distinct neurons in a central structure receive coincident input from both ears only when sounds have particular, distinct onset differences and thus are likely to emerge from a particular point in space[1, 2].

However, the link between structure and function is not always so direct. In particular, mammalian cortex is thought to be highly modular[3–6]. This modularity in structure is suggested by cytoarchitectural differences between different areas[3, 4] and anatomical tracing of connections between those areas[5–8]. Further, modularity in function is suggested by electrophysiological recordings within different areas (e.g., [8–11] and many more) and lesion experiments, in which specific behavioral deficits are observed after different areas are damaged or removed (e.g., [12–16] and many more). However, the direct functional consequences of this modularity have proven difficult to fully understand.

As an example, the primate visual system is understood to be elaborately modular in both

structure and function (Figure 1.1a)[5, 6]. One example of this modularity comes from the two stream hypothesis (Figure 1.1b)[17–20]. This idea, now over half a century old, splits the primate visual system into two anatomically and functionally distinct processing hierarchies, each of which is composed of many cortical regions. The ventral visual stream – also referred to as the "what" or vision-for-perception stream – is thought to be specialized for extracting complex, detailed representations of the component objects of visual scenes. This stream culminates in the inferotemporal cortex (ITC), where single neurons have been shown to exhibit rich representations of conspecific (and related species) faces[21] as well as highly invariant[22, 23] representations of objects. The second visual processing stream, the dorsal stream – also referred to as the "where" or vision-for-action stream – is thought to be specialized for the planning of visually guided actions[13, 24], spatial representations[25], and the deployment of spatial attention[26]. This stream culminates in the posterior parietal cortex, which has a rich representation of motor intention[26] and which is thought to implement a spatial priority map[27]. That is, a topographic map of sensory space where the neural activity at a point in the map reflects the behavioral relevance of that location in space – and, thus, the likelihood that the animal will deploy attention to that location.

This dichotomy has profoundly shaped our understanding of the primate visual system, both by providing an organizing framework but also by shaping the kinds of experiments that are performed. In particular, while there are hundreds of studies that elaborate on how individual brain regions contribute to the function of the processing stream that they are located within, relatively little is understood about the role those brain regions have in functions that are thought to belong to the complementary processing stream. Yet, when such function is investigated directly it is often discovered – as with rich three-dimensional object representations[29, 30] and shape representations[31] in the dorsal stream. This richness of function in single brain regions suggests that the brain's modularity may, in fact, be weaker than strong dichotomies, like the two streams hypothesis, suggest. Further, theoretical work

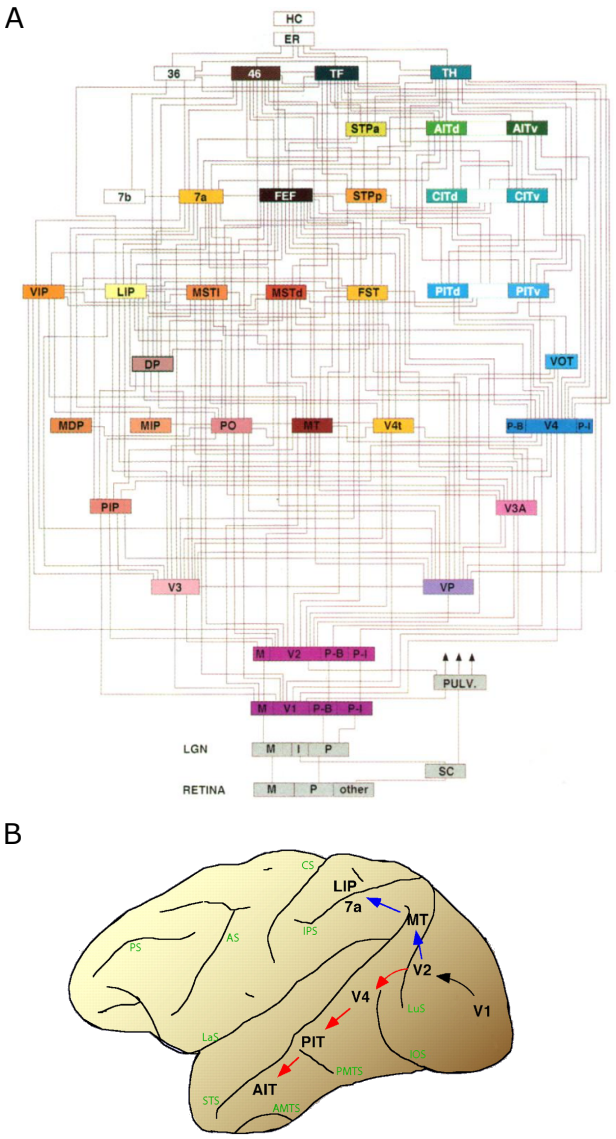

Figure 1.1: Hierarchies of visual cortical areas.
A The van Essen diagram depicts the primate visual system as a modular system of interconnected brain regions. Adapted from [5, 6]. **B** The two-streams dichotomy takes a more explicitly functional view of a similar set of brain regions, dividing them into dorsal and ventral regions, each with distinct functional specialization. The ventral stream (in red) for object recognition; and the dorsal stream (in blue) for spatial processing and visually guided actions. Adapted from [28].

described in chapter 3 demonstrates that strong modularity, where two brain regions represent completely distinct information about the sensory world, is not a computationally feasible strategy in rich environments with multiple distinct objects. In general, precisely

characterizing the function of distinct cortical regions has proven difficult. In part, this is because experiments are limited to detecting only the limited set of functions that they explicitly test for – and because inferences made from these finite observations often fail to generalize (see chapter 4).

Further, recent experimental work has shown that the causal experimental techniques often used to establish the unique function of different brain regions are more difficult to interpret than previously thought[32]. In particular, to test whether a region is necessary for a particular behavioral function, common practice is to either chronically ablate[12–16] or acutely inactivate[33, 34] the region of interest, and then perform behavioral testing during the manipulation. However, chronic and acute causal manipulations produce biases in opposite directions. Chronic ablations often underestimate the effects of removal, as the animals typically need time to recover from the procedure and they are able to learn to compensate for the deficit in that interim[32]. On the other hand, acute inactivations often overestimate the importance of a region to a particular behavior[32]. This is because the abrupt silencing of a particular region can cause unanticipated perturbations in the dynamics of downstream brain regions, through cross-region interactions[35, 36]. Thus, while anatomically distinct brain regions have been conclusively demonstrated, the precise function of these regions is often difficult to isolate relative to the function of a collection of regions. That is, a mapping between the structural and functional modularity in cortex has remained elusive.

One difficulty in understanding the functional modularity of the brain may arise from a lack of precision in language about neural computations and representations more generally. In particular, if a brain region is found to represent a particular quantity, it is often then inferred that the brain region has some role in computing that quantity. We believe that sharpening the delineation between representation and computation may help to resolve ambiguity in functional modularity. As an example, there is compelling evidence that complex and invari-

ant object representations are computed from natural images across the hierarchy of regions in the primate ventral visual stream[37, 38]. Further, the representations at each stage in the hierarchy are well-matched by a feedforward deep neural network trained to perform object recognition[39, 40]. This observation implies that much of the computation occurs in the communication between brain regions, rather than through recurrent processing within brain regions. That is, while each region has progressively more refined object representations, the bulk of the computation occurs outside of any particular region. Thus, it is necessary to compare neural representations a step apart from each other in the processing hierarchy to make inferences about the computation itself. As an example, in the ventral stream, experimental and theoretical work has demonstrated that object representations emerge, in part, through the progressive untangling of object identity information from general visual information (e.g., this untangling process would cause distinct objects with similar low-level features to be represented increasingly differently) in transitions between V4 and ITC[37] as well as ITC and perirhinal cortex[41]. While the primate ventral visual stream may be particularly well-suited to this explicit understanding of computation, the recent move toward understanding neural population activity in terms of dynamics on a manifold may help clarify which computations happen across regions, as expressed by deformations between the manifolds in two distinct brain regions[42, 43], and which happen within a region, as expressed by particular dynamics on the manifold in that region[44, 45].

To move toward this understanding, it is important to understand the constraints on representations and dynamics within each region, constraints on how distinct regions can interact with each other, as well as how these interactions depend on distinct behaviors. First, constraints on representations and dynamics within single regions have been studied extensively in the literature on neural coding. We will review how neural representations can achieve the three normative goals of reliability in the face of neural noise and cell death, efficiency in terms of both population spiking activity and the number of neurons required in the

population, and accessibility of both general and specific information to downstream neural populations. Second, the interactions between distinct populations have received relatively little attention in theoretical and experimental neuroscience. We will review constraints on neural dynamics that can be produced by interacting brain regions and recent theoretical work that describes constraints on the information represented by interacting brain regions. Third, neural activity is often surprisingly context-dependent. While one might expect brain regions in a modular system to perform a single computation in the same way across all behavioral contexts, experimental work has illustrated that this is often not the case. We will review experiments demonstrating this phenomenon and connect it to difficulties in understanding the brain as a modular system. Finally, we will briefly discuss the integration of these three threads into a unified understanding of distributed computation in the brain.

1.2 Reliable, efficient, and accessible neural representations

The theoretical study of neural representations has followed the normative principles of the efficient coding hypothesis[46, 47] for more than half a century. This hypothesis suggests that neurons are constrained to represent the sensory environment efficiently; that is, a neuron should communicate as much unique information about the sensory environment as possible while using as few energetically expensive spikes as possible. Relatedly, this also means that neurons should seek to transmit as little redundant information about the environment as possible. In practice, this redundancy reduction aspect of efficient coding can be achieved by learning the statistical structure of the sensory environment. Then, for instance, if two phenomena are highly correlated with each other, redundancy can be reduced by transmitting only one of them. This relatively straightforward principle of efficient coding through redundancy reduction can be made concrete using the mathematics of information theory. With that formalization, efficient coding has proven effective at predicting the structure of neural codes, especially on the sensory periphery[48–50] and in early sensory areas[51, 52].

However, efficient and wholly non-redundant codes also have drawbacks[47]. In particular, redundancy is essential to constructing reliable codes – as, in the presence of significant noise, it is precisely redundant information that allows noise to be disentangled from signal, and corrected. Further, while single neurons in isolation can be highly reliable, neurons in cortex are often highly variable in their activity[53, 54], potentially due to their heterogeneous input that reflects a wide variety of sensory, contextual, and motor features[55, 56]. Thus, overcoming noise to produce reliable neural representations is a significant challenge for cortical and some subcortical representations. Combining the ideas of efficient and reliable coding have also been successful in predicting the structure of some neural representations. In particular, grid cells in the entorhinal cortex can be viewed as implementing an exquisitely reliable analog code for spatial position[57] and the high-dimensional representations of sensory, cognitive, and motor variables observed in cortex[58] have also been shown to implement a reliable and efficient code (see chapter 2 and [59]). These results indicate that reliability, in addition to efficiency, is an important constraint on the structure of neural representations.

Information theory addresses the dual concerns of efficiency and reliability, and provides theoretical limits on the level of reliability that can be achieved given a particular amount of energy. However, one drawback of many codes that achieve theoretically optimal reliability and efficiency is in decoding. In coding theory as studied for telecommunications, the decoding step for optimal codes often requires a sequence of complex operations to be performed on the received information before all of the theoretically correctable errors introduced during transmission are corrected in practice[60, 61]. This poses a problem for neural implementations of these codes, as the implementation of similar operations would require recurrent processing that is inconsistent with the rapid feedforward transmission of information through hierarchical neural systems, such as the ventral visual stream[37, 38]. This suggests a third constraint on neural representations: They must format information in a way that is accessible for rapid error correction and reformatting.

This accessibility constraint has been most clearly described in work on a common form of neural selectivity known as nonlinear mixed selectivity[62, 63]. In these mixed codes, the response of a single neuron is nonlinearly modulated by multiple sensory, cognitive, and motor features – in particular, mixed selective neurons typically respond strongly to one or more combinations of these feature values, but are unresponsive to the constituent features alone. This nonlinear conjunction operation has the consequence of expanding the dimensionality of the representation of all of the stimulus features, and simple linear decoders can then be used to decode any individual stimulus or any combination of stimuli. Thus, mixed selectivity can support rapid and flexible decoding. These mixed codes have been shown to be implemented across cortex, including in sensory[64–69], frontal[62, 63], and motor cortices[70–72]. Further, recent work[59] (and see chapter 2) has demonstrated that mixed codes are also more reliable and efficient than non-mixed codes – though less reliable and efficient than optimal codes. Further, this work showed that mixed codes provide these accessibility, reliability, and efficiency benefits in the sensory context as well as for mixtures of higher-level stimulus features as was previously the focus. Thus, mixed representations may provide a compromise between the necessity of rapid decoding and reformatting as well as reliability and efficiency.

However, mixed representations alone also have an important drawback. While the high-dimensional representations produced by mixed selectivity are beneficial to accessibility, efficiency, and reliability as described above, they can be detrimental to learning and generalization[59, 73, 74]. As an example, the representations of a pair of stimuli that are close to each other in stimulus space will become less and less similar to each other as the degree of mixing in the code increases. Thus, it can become impossible to infer that the two stimuli are similar to each other from their neural representations alone. Recent work has shown that neural representations in the prefrontal cortex implement a compromise between highly mixed representations and low-dimensional structure that can be used to support gen-

eralization and abstraction[75]. A similar implementation of a high-dimensional code with low-dimensional structure has recently been observed for representations of natural images in mouse visual cortex as well[58]. Further, unpublished work from the author has shown that these heterogeneous mixed and non-mixed combination codes preserve many of the efficiency and reliability benefits produced by mixed codes alone.

1.2.1 The geometry of reliable, efficient, and accessible codes

All of these constraints on neural codes can be expressed in a unified way through the code geometry. That is, each stimulus that is represented by a code will correspond to a point in representation space. Each neuron in the code population corresponds to an axis of this representation space. The metabolic cost of spiking activity in the code will be related to the average distance from the origin of each stimulus representation. Similarly, the reliability of the code is related to the distance between the representations of two stimuli that are nearby to each other in stimulus space. As an example, one way to increase the reliability of the code is to increase the gain of every neuron in the population – that is, where a neuron fired n spikes to a particular stimulus before, it will fire $2n$ spikes to the same stimulus now. This gain increase in spiking necessarily scales the distance between all stimulus pairs by a factor of 2 as well. Thus, two things happen: First, the code becomes more reliable because all of the stimuli are now further apart; second, the metabolic cost increases as all of the neurons now have higher firing rates on average. In a similar context, coding theory has illustrated that this mechanism of increasing code reliability is highly inefficient[60].

Instead, manipulation of the geometry of the representation space itself can increase reliability without decreasing efficiency. In the case of nonlinear mixed selectivity, the increased reliability arises from the expanded dimensionality of the representations (Figure 1.2). Even if all stimulus representations are constrained to be a fixed distance from the origin, a high-dimensional representation will have more ways to arrange the representations such that

they are further apart, while a lower-dimensional representation will be forced to place the representations closer together. While this is perhaps easiest to conceptualize with a discrete set of points, this is true for a continuous manifold as well. To illustrate this, consider the extreme case where each stimulus is represented on its own dimension in the high-dimensional code (Figure 1.2left). For positive firing rates and a fixed distance from the origin, every stimulus will be at the largest possible distance from every other stimulus. Thus, any lower-dimensional code will be forced to place at least one stimulus at a distance that is at least slightly smaller than that maximum (Figure 1.2right)[59].

On the other hand, it is precisely this uniformly maximal distance that can make high-dimensional codes difficult to learn. To understand why this is, we again return to our extreme example where each stimulus is represented along its own dimension. Here, because all stimuli are at the same distance from each other, we have destroyed any information about whether two stimuli are similar – either in the sense of being nearby to each other in stimulus space or sharing a subset of feature values. Thus, anything that we want to learn about a particular feature value (say, a single feature value predicts reward) must be learned about each stimulus with that feature value individually. To ameliorate this difficulty, neural codes can incorporate low-dimensional structure that provides this information, while still maintaining high-dimensional representations. This can be done both by increasing receptive field size[59] as well as by including non-mixed terms in the representation[75]. Both strategies decrease reliability and efficiency from the maximum achievable by mixed selectivity, but still retain benefits over non-mixed representations.

Thus, reliable and efficient codes that provide accessible and learnable information have a distinct geometry: They should be high-dimensional to provide accessible, efficient, and reliable representations, but they should be mixed in with relatively low-dimensional structure to provide information useful for learning abstractions and generalizing across distinct

contexts. Recent experimental and theoretical work has indicated that both the monkey prefrontal[62, 63, 75] and parietal[59] cortices and the mouse visual cortex[58] exhibit codes with this characteristic mixture of high- and low-dimensional structure. In the mouse visual cortex, it was shown that the representations of natural images appear to lie at the optimal point on a trade-off between that high- and low-dimensionality[58].

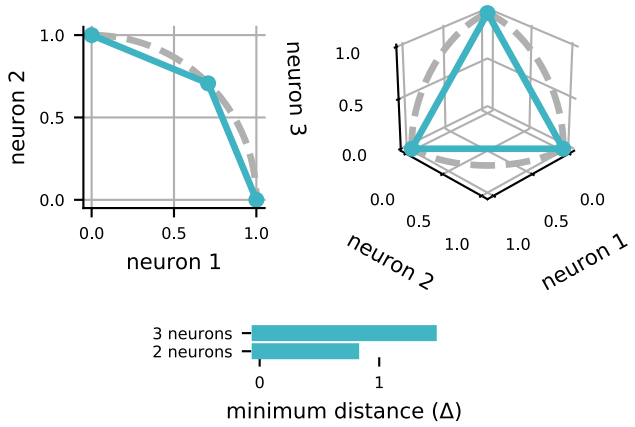

Figure 1.2: Differences in representation separation between low- and high-dimensional neural codes. (left) A low dimensional representation of three stimuli, constrained to exhibit representations with fixed distance from the origin. (right) A representation of the same three stimuli with expanded dimensionality. (bottom) The smallest distance between any pair of representations in each code. This minimum distance provides a good approximation of code performance.

1.3 Neural representations in a distributed system

While normative theories of neural representations – such as efficient coding – produce constraints on neural representations within a single brain region, the modular nature of the brain implies that there is an additional set of constraints on representations across interacting brain regions. First, structural modularity implies constraints on the dynamics in individual brain regions due to their placement within a hierarchy of interacting brain regions; second, functional modularity implies constraints on the information represented within each brain region, such that interactions between regions can be reliable. These dynamical and information constraints have yet to be integrated with one another. Further,

a comprehensive theory of neural coding that integrates both the constraints on within region representations discussed in the previous section and the constraints implied by the brain's modularity has yet to be developed.

The dynamics of neural activity within a region have long been understood as being primarily the result of local circuit properties. In particular, phenomena such as divisive normalization[76] and diffusion-to-bound decision dynamics[77] have been shown to be well-explained by interactions between neurons within a single brain region. Even the sustained activity that is thought to support working memory has been explained in terms of local network interactions[78, 79] (though see [36]). However, each of these explanations assume that the region being studied receives an input relevant to its function and produces an output. In the brain, the situation is much more complicated. Not only do neural populations tend to receive extremely heterogeneous inputs[56, 58], but the activity in one region affects the activity in connected regions. The activity in these connected regions is then, in some cases almost immediately, fed back into the original region – thus, there are additional inter-region recurrent loops that can strongly affect the dynamics within any particular region. It is unclear what the role of this larger scale of recurrence is for computation.

1.3.1 Constraints on neural dynamics from distinct brain regions

The broad dynamic effects of this recurrence has been studied using a large-scale model of monkey cortex, where the dynamics of 29 interconnected brain areas were modeled using experimentally determined functional connectivity weights[35]. The most striking effect of this contextualization of single region dynamics within dozens of interconnected regions is an increase in the time constant of neural dynamics in regions that are located increasingly higher in the hierarchy, even given neuron models with fixed time constants in every region[35]. This means that regions that are higher in the cortical hierarchy, such as regions in prefrontal cortex and at the apex of the dorsal and ventral visual streams (in posterior pari-

etal cortex and the inferotemporal cortex, respectively) tend to integrate inputs over much longer timescales and have dynamics that reflect inputs from more distant points in the past than primarily sensory regions that are lower in the hierarchy (Figure 1.3). In addition, areas that were higher in the hierarchy, were also more likely to exhibit stable persistent activity[35], which is viewed as a hallmark of more cognitive computations.

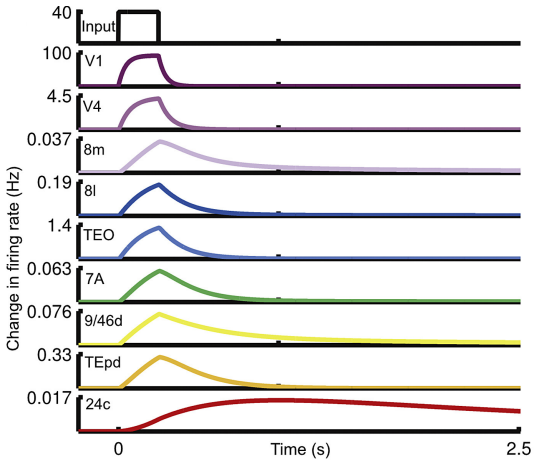

Figure 1.3: Each different colored trace is the response of a brain region to pulsed input (top). The duration of the response indicates the time constant of the dynamics in that brain region. Brain regions lower on the plot are higher in the cortical hierarchy, and have properties matched to the properties of that region, including connection weights to the other region. Adapted from [35].

Further investigation of a similar model reveals that the extensive feedback and recurrent loop projections present in cortex can destabilize signal propagation from lower to higher regions in the hierarchy[80]. In particular, these dynamics appear to force the network of regions into one of two states: a state of extreme signal attenuation, in which signals from lower regions in the hierarchy are scarcely able to reach higher regions, and a state of global instability, where inputs to lower regions can produce massive global activations that are coupled across all the regions in the network[80]. Both regimes are unrealistic. To achieve reliable signal transmission across the network of brain regions, excitatory coupling between the regions and inhibitory coupling within each region was increased[80]. This manipulation also had the effect of creating a threshold effect on transmission – in which inputs to lower sensory

areas are only represented in parietal and frontal areas if they are sufficiently strong[80]. This filtering mechanism could not be inferred from the study of single brain regions in isolation. The network of regions explored in these two modeling studies provide non-trivial constraints on population dynamics both within and across single regions. To develop a comprehensive understanding of neural computation, it is important to incorporate these constraints.

1.3.2 Constraints on neural representations from distinct brain regions

While the general constraints on dynamics discussed above fundamentally shape interactions between brain regions, they give relatively little insight into the specifics of information integration and computation performed between those regions – particularly in the coding theoretic sense of investigating efficiency, reliability, and accessibility of neural representations across regions. Information integration across multiple regions has received the most attention in the context of multisensory integration. In particular, there is an extensive literature studying the estimation of heading direction from combined visual and vestibular signals[81, 82]. This work indicates that neural systems optimally combine the two sources of information to arrive at a more precise estimate of heading direction than could be derived from just one of the forms of information alone[81].

Further, modeling work has developed a recurrent neural network system that performs both optimal cue integration and provides a solution to the so-called recoding problem[83]. The recoding problem concerns how representations of the same quantity that are encoded in different frames of reference can be integrated with one another. As an example, the auditory system estimates spatial position in a head-centered frame of reference, while the visual system estimates spatial position in an eye-centered frame of reference. Given information about eye position, these two estimates of spatial position can be integrated with one another, and an estimate that leverages this combined information will be more accurate than estimates that rely on either individually (just as with visual and vestibular integration above). In this

work, an intermediate population of neurons is hypothesized to simultaneously recode from eye-centered to head-centered and from head-centered to eye-centered reference frames, given designed connectivity between the three populations of neurons representing eye-centered position, head-centered position, and eye position. Through recurrent dynamics between these four total populations, the estimates in each population are iteratively refined and converge toward the optimal estimate in each domain[83]. Interestingly, this integration strategy does not assume a dominant frame of reference. Instead, optimal estimates are developed within both head- and eye- frames of reference. Further, the model predicts that the intermediate population that mediates the integration process will exhibit shifting patterns of selectivity according to changes in the eye position, similar to those observed in the posterior parietal cortex[84, 85]. Thus, this work provides a neurally plausible and experimentally supported mechanism for solving the recoding problem.

However, the previously discussed work and almost all theoretical work on multisensory integration in general supposes that there is only a single stimulus representation within each modality, and that the representations in each modality are guaranteed to emerge from the same source in the world. In visual and vestibular integration, this is true for static visual environments where the only motion present corresponds to self motion, but it is not true in natural environments, which often have many moving components. Behaviorally, both humans[86] and animals[87] have been shown to integrate vestibular and visual self motion cues in the presence of distractor object motion. While humans have been shown to perform this integration near optimally using a causal inference framework[86], the neural implementation of this computation remains unclear.

Further, in other multisensory integration contexts, there is no guarantee that integration should proceed at all. As an example, when presented with a single auditory stimulus in conjunction with a single visual stimulus, two possible scenarios can be true. First, both

stimuli can arise from the same source – that is, the same cause in the world is responsible for both stimuli. Second, each stimulus can arise from a different source – that is, there are two distinct causes, one for each of the stimuli. Discriminating between these two scenarios is highly non-trivial, though humans are, again, shown to do so near-optimally according to a Bayesian inference framework[88]. However, this integration process becomes even more complicated in the presence of additional stimuli. Even if we make the simplifying assumption that each stimulus in one modality will necessarily have the same cause as one (and only one) stimulus in the other modality, the number of possible integrations grows as the factorial of the number of stimuli. The integration of distinct sets of representations is referred to as the assignment problem[89]. This assignment problem must be solved when integrating any sets of representations, not only representations that arise from distinct modalities. Thus, insight into necessary conditions for a solution to the assignment problem will also give insight into constraints on neural representations across brain regions.

The mechanisms of the general assignment problem are relatively understudied. While there is an extensive literature in human psychophysics on feature integration for object segmentation[90, 91], there is relatively little quantitative modeling that addresses the integration of arbitrary numbers of objects. Recent work (see chapter 3) generalizes the insight that integration processes rely on shared representations of at least one object feature – in the cases discussed so far, this is a shared representation of position – and describes how the assignment error rate depends on changes to this shared information. This framework describes a quantitative trade-off between increasing the level of redundant information represented between two brain regions, which leads to a decrease in the assignment error rate, and decreasing the level of redundant information, which allows all of the information to be represented with greater accuracy. However, crucially, representations across distinct processing hierarchies cannot be mutually independent, as the efficient coding hypothesis would suggest. Redundancy between representations of the sensory world is strictly necessary for disambiguating

which objects (or causes) in the world are giving rise to which sensory phenomena. This constraint on neural representations across brain regions may be the reason that modularity in the brain is not as strict as originally suggested by strong dichotomies, such as in a strong interpretation of the two streams hypothesis for the primate visual system.

To understand the solution to the assignment problem, a geometric perspective is again helpful. Each of two, for instance, sensory brain regions represents K_1 and K_2 object features. That is, for N stimuli, each brain region represents N points in a K_1- or K_2-dimensional space. For a reliable solution to the assignment problem to exist, some of these dimensions must be overlapping. This overlap does not need to be explicit. For instance, if two features in the first region are correlated with one of the features in the second region, that is enough. The set of N objects represented in each region can then be projected into this lower-dimensional, common representation space. In this framework, the solution to the assignment problem is a mapping between the two sets of representations in the common representation space. In chapter 3, we show that the optimal mapping is the one that minimizes the sum squared distance between the integrated representation pairs. Thus, we can see that assignment errors are most likely to occur when clusters of stimuli are nearby to each other in this common feature space. In human psychophysical data where assignment errors have been demonstrated, these errors are indeed most likely[92]. While a neural implementation of this solution to the assignment problem is straightforward using the ingredients developed previously in the literature and described above, there are interesting questions about how dynamics specific to multiple object representations, such as divisive normalization, affect the assumption that the estimates of each object are independent from each other, and how this depends on the geometry of the neural representations in each individual region.

There is still work to be done toward a unified theory of distributed neural representations. However, the above work provides key ingredients to this theory by outlining constraints on

neural dynamics and information representation in a weakly modular neural system. The mutual interaction between these constraints and the geometry of neural representations in a single brain region remain to be determined. Further, so far, we have elided any discussion of the rich behavior that these neural representations and dynamics produce. Recent work has illustrated difficulties in understanding the function of different brain regions from limited behavioral probes. In particular, neural representations in different brain regions appear to be richly context-dependent, which complicates attempts to succinctly describe their function. Thus, a unified theory of distributed neural representations must account for this dynamic as well.

1.4 Context-dependent coding

Behavioral context has been shown to significantly shape neural activity, sometimes in surprising ways. The behavioral contexts that can be explored in a laboratory setting are often limited, and occupy a relatively unnatural subspace of all possible behaviors[93]. While strictly controlled experiments with constrained and unnatural stimuli can give fundamental insight into information processing in the brain, it is important to then test these conclusions with both more natural stimuli[94] and in additional, potentially more naturalistic, behavioral contexts. As an example, activity in some regions of the songbird song-production system, activity is highly dependent on the animal's social context[95]. In particular, when the animal is singing in the presence of a conspecific as opposed to singing alone, the activity in several nuclei involved in song production is markedly reduced and markedly less variable[95]. This contrast in behavioral contexts provides unique insight into the function of these brain regions for song production: They indicate that the two nuclei are preferentially involved in song learning during private rehearsal[96].

Behavioral context has also been shown to be deeply important to cortical activity in the

mouse as well. When animals are running, the responsiveness of visual cortical neurons typically doubles relative to when the animal is still – though the stimulus tuning of those responses remains similar[97]. This change has been shown to increase the signal-to-noise ratio of visual representations[98], and may reflect an overall increase in attention to visual inputs. One interpretation of this result is that the mouse weights visual information more heavily while ambulating and weights it less while staying still – perhaps giving priority, instead, to other sensory information. In this case again behavioral context is revealed to have a crucial role in determining neural function.

Further, recent work in the monkey (see chapter 4) has shown that the presence of explicit task demands – referred to as a directed task – strongly modulates representations of behaviorally relevant information in the posterior parietal cortex (PPC) relative to a task without explicit demands, but during which the animal is making identical behavioral responses. These results further highlight the importance of accounting for task context when making inferences about the function of different brain regions. In particular, previous work has suggested that the lateral intraparietal area (LIP) in the PPC has a general role in the allocation of bottom-up spatial attention[99] and in the representation of a generic map of salience and behavioral priority in the sensory environment[27]. These recent results instead suggest that LIP may play this role in the allocation of spatial attention only in directed task contexts with explicit reward structure.

All of these examples illustrate that our understanding of neural function is highly conditioned on the behavioral context of the animal, sometimes in surprising ways. Further, results indicating that LIP may play a distinct role in directed relative to undirected tasks suggests that modularity in the brain may, in some cases, be more directly organized around particular behaviors rather than specific computational functions. For instance, LIP may not have a general role in computing the bottom-up salience (or priority) of a particular

stimulus; instead, it might be specialized for the rapid selection of cued visual stimuli, and represents the salience of visual stimuli only when it is relevant to supporting that behavior. Theoretical work that explores the trade-off between modules that perform a specific and generic computation rather than modules that are specialized to support specific sets of behaviors is a necessary ingredient for any unified theory of distributed computation in the brain.

1.5 Conclusions and remarks on the following chapters

We have outlined three areas where significant progress has been achieved in understanding neural function: representations within brain regions, the integration of representations from across distinct brain regions, and the often striking dependence of neural activity on behavioral context. Throughout, we have highlighted the significant challenges that still remain in unifying these three to-now disparate fields of inquiry into a more comprehensive understanding of neural function as a distributed computational process. While significant progress has been made in understanding the construction of efficient, reliable, and accessible population codes within a single brain region, further work is necessary to understand how these population codes can interact with each other outside of strict feedforward interactions and beyond cases with only a single stimulus representation. The brain effortlessly represents numerous multisensory stimuli at once and across time – yet, the neural mechanisms of these heterogeneous and dynamic representations have received surprisingly little attention.

Further, our understanding of the functional role of distinct brain regions is severely impoverished. While a single experiment can provide information about a region's involvement in a particular task or set of tasks, it has proven difficult to make reliable inferences about the general function of brain regions from these relatively isolated observations. We suggest that our understanding of functional modularity would be improved by developing a stronger

theoretical basis for distributed computation in systems like the brain. In particular, it may be productive to reframe the function of some brain regions more explicitly in terms of the behaviors that they mediate rather than searching for a specific and generic computational function that they serve.

In the following chapters, we expand on each of these areas of research in original work, that has also been referenced within each section. First, we investigate the production of reliable, efficient, and accessible population codes in cortex and show that nonlinear mixed selectivity is a desirable coding strategy for achieving each of these goals. Second, we investigate the integration of distinct representations of multiple objects by extending classic optimal integration frameworks to include multiple stimuli. Here, we show that a balance of redundancy and uniqueness in neural representations across brain areas is necessary to navigate a trade-off between catastrophic integration errors and the fidelity of neural representations generally. Finally, in the third chapter, we investigate context dependence in the posterior parietal cortex area LIP using two distinct behavioral tasks: One that requires directed stimulus selection and the other that allows the animal to select stimuli according to its innate preference. This experimental work reveals that LIP is preferentially engaged in the directed relative to the undirected behavior – again, revealing the often fundamental context dependence of neural representation in many brain regions.

CHAPTER 2

NONLINEAR MIXED SELECTIVITY SUPPORTS RELIABLE NEURAL COMPUTATION

W. Jeffrey Johnston, Stephanie E. Palmer, and David J. Freedman

now published in *PLoS Computational Biology*[59]

Abstract

Neuronal activity in the brain is variable, yet both perception and behavior are generally reliable. How does the brain achieve this? Here, we show that the conjunctive coding of multiple stimulus features, commonly known as nonlinear mixed selectivity, may be used by the brain to support reliable information transmission using unreliable neurons. Nonlinearly mixed feature representations have been observed throughout primary sensory, decision-making, and motor brain areas. In these areas, different features are almost always nonlinearly mixed to some degree, rather than represented separately or with only additive (linear) mixing, which we refer to as pure selectivity. Mixed selectivity has been previously shown to support flexible linear decoding for complex behavioral tasks. Here, we show that it has another important benefit: in many cases, it makes orders of magnitude fewer decoding errors than pure selectivity even when both forms of selectivity use the same number of spikes. This benefit holds for sensory, motor, and more abstract, cognitive representations. Further, we show experimental evidence that mixed selectivity exists in the brain even when it does not enable behaviorally useful linear decoding. This suggests that nonlinear mixed selectivity may be a general coding scheme exploited by the brain for reliable and efficient neural computation.

2.1 Author summary

Neurons in the brain are unreliable, while both perception and behavior are generally reliable. In this work, we study how the neural population response to sensory, motor, and cognitive features can produce this reliability. Across the brain, single neurons have been shown to respond to particular conjunctions of multiple features, termed nonlinear mixed selectivity. In this work, we show that populations of these mixed selective neurons lead to many fewer decoding errors than populations without mixed selectivity, even when both neural codes are given the same number of spikes. We show that the reliability benefits from mixed selectivity are quite general, holding under different assumptions about metabolic costs and neural noise as well as for both categorical and sensory errors. Further, previous theoretical work has shown that mixed selectivity enables the learning of complex behaviors with simple decoders. Through the analysis of neural data, we show that the brain implements mixed selectivity even when it would not serve this purpose. Thus, we argue that the brain also implements mixed selectivity to exploit its general benefits for reliable and efficient neural computation.

Keywords: neural coding, reliability, efficiency, nonlinear mixed selectivity, conjunctive coding

2.2 Introduction

To support behavior, the brain must use a communication strategy that transmits information about the world faithfully, efficiently, and, perhaps most of all, reliably. The first two of these goals have received extensive attention in neuroscience, particularly in the literature on efficient coding and redundancy reduction[46]. Efficient coding focuses on discovering the response field (RF) for a single neuron that simultaneously maximizes the amount of stimulus information transmitted by the neuron while minimizing the number of spikes that

the neuron must fire[46]. A crucial step to this process is representing stimuli without any of the redundancy inherent to the natural world – that is, by isolating and representing the independent components of natural stimuli[100]. Refinements of efficient coding[47] have also emphasized the need for the representation of these components to be neatly packaged, or formatted, so that they are accessible to decoding (as with nonlinear mixed selectivity[63]) and facilitate generalization[101]. As a whole, the ideas of efficient coding have been used to accurately predict the structure of RFs in primary visual cortex[51, 102], and other sensory systems[103–105]. However, existing work on efficient coding, redundancy reduction, and neat packaging primarily addresses the goals of faithful representation and metabolic efficiency. This work does not typically characterize the reliability of decoding after these efficient representations are corrupted by the noise that is inherent to single neuron responses[53, 54]. In fact, non-redundant representations are often highly vulnerable to noise[60].

Making efficient representations robust to the noise present throughout neural systems has received considerably less attention in neuroscience. In information theory, noise robustness is the goal of channel coding. The channel code re-encodes efficient and non-redundant stimulus representations to include redundancy that will increase the robustness of that stimulus representation to later corruption by noise. Recent work has shown that grid cell RFs[57] and the working memory system[106] may implement near-optimal channel codes. In sensory systems, channel coding has been explored more obliquely. Extensive work has focused on deriving RF properties that maximize mutual information between the stimulus and the response[107–110] or the Fisher information from the response function[111–113] (and see [114, 115] for connections between these approaches). However, these measures do not always imply a particular level of decoder performance. Mutual information connects to decoder performance via the rate-distortion bound[61], but different codes with the same mutual information can have different levels of decoder performance relative to that bound[116] (and see Figure 2.8B). Further, the kind of information encoded matters: a code

that has lots of information about a target stimulus without information about which stimuli are nearby to that target will minimize the probability of decoding error, but have worse performance on distance-based measures of error because its errors will be random with respect to the original target; a different code with the same amount of mutual information may make the opposite tradeoff, and minimize distance-based errors while increasing the overall frequency of errors. Evaluating mutual information for each code will not indicate which kind of errors it is likely to make – here, we explore the tradeoff between these two kinds of errors explicitly (see Figure 2.3). Finally, Fisher information is linked to decoder performance via the Cramer-Rao bound, but saturation of this bound is only guaranteed in low-noise conditions[117] (and codes with less Fisher information can outperform codes with more Fisher information when optimal decoding cannot achieve the bound[118]). There is neural and behavioral[119] evidence that the brain computes successfully on short (e.g., $\sim$80 ms) timescales and spiking responses have been shown to be highly variable on that timescale[119], thus it is unlikely that the brain typically operates in a low-noise regime.

Here, we analyze an ubiquitous coding strategy in the brain – conjunctive coding for multiple stimulus features – in terms of both its reliability and efficiency. Previous work on conjunctive coding (commonly called nonlinear mixed selectivity[62, 63]) has shown that it produces a neatly packaged and sparse representation that enables the use of simple linear decoders for complex cognitive tasks[63], particularly in the macaque prefrontal cortex[62]. Further, random conjunctive coding has been shown to increase the number of discrete stimuli that can be reliably represented in a neural population[120, 121], particularly in the context of the olfactory system[122–124]; however, a detailed analysis of how the error rate of these codes depends on metabolic cost was not performed. In our work, we develop a novel generalization of nonlinear mixed selectivity, allowing different levels of mixing between stimulus features while preserving full coverage of the stimulus space (see *Definition of the codes* in *Methods*). Using these codes, we show that the encoding of stimuli with at least some

level of nonlinear mixing almost always produces more reliable and efficient communication than without mixing. Further, we demonstrate novel tradeoffs between codes with and without mixed selectivity – including an analysis of how RF size and error-type affect the optimal level of mixing. Finally, we link our work to experimental data by showing that mixed selectivity is implemented in the brain even when it does not support the flexible linear decoding of stimulus features, but would still play a role in improving the overall reliability of decoding. Our work illustrates that nonlinear mixed selectivity provides highly general benefits to coding reliability and efficiency, and helps to explain the ubiquity of mixed selectivity within sensory[64–69], frontal[62, 63], and motor cortices[70–72].

2.3 Results

2.3.1 Increased mixing increases stimulus discriminability

In the brain, stimulus representations are corrupted by noise as they are transmitted between different neural populations. This process can be formalized as transmission down a noisy channel (Figure 2.1A). The reliability and efficiency of these transmissions depends on the format of the encoded representations – here, we show how three different properties of this representation are affected by nonlinear mixing, and how those properties interact with transmission reliability and efficiency. The three properties of neural representations that we focus on are: minimum distance, neural population size, and metabolic representation energy (Figure 2.1B and *Code properties* in *Methods*).

Minimum distance is the distance between the representations of the two stimuli that are most difficult to discriminate – i.e., that have the most similar neural responses. Importantly, half of this minimum distance represents the smallest magnitude of noise that could cause a decoding error given optimal decoding. Since smaller noise perturbations are more likely to occur than larger ones, errors that map the response to one of the stimuli at minimum

distance are more likely than errors to any other stimuli. As a consequence of this, a larger minimum distance typically implies a lower overall probability of error and the minimum distance can be used to develop an accurate approximation of the overall probability of error in many conditions. We develop both a minimum distance-based approximation and an approximation based on the likelihood of all possible errors in *Estimating the error rate* in *Methods*. In general, the minimum distance is a more useful metric for summarizing our codes than, for instance, the average distance, because the error rate is a nonlinear function of the distances between individual stimuli. As an example, a code with half of its stimuli at a small minimum distance and the other half of its stimuli at a much further maximum distance would, in most cases, have a much higher error rate than a code that has all of its stimuli at the average of the near and far distances. Population size is the minimum number of independent coding units, or neurons, required to implement the code such that all possible stimuli have a unique response pattern. Representation energy is the metabolic energy consumed by the response of the neural population to a stimulus, defined as the square of the distance between the zero-activity state and the response patterns to stimuli for the code – here, representation energy can be viewed as the squared spike rate in response to a particular stimulus summed across the population of neurons used by the code (though we also consider the sum spike rate, see Figure 2.5). In the codes we consider here, all of the stimuli evoke the same number of spikes across the population, and therefore have the same representation energy. Representation energy represents the active, metabolic cost of the code (in terms of the cost of emitting spikes), while population size represents the passive metabolic cost of the code (in terms of neuronal maintenance costs across the population, spiking or not). We begin by considering representation energy alone before considering both together.

The stimuli represented by our codes are described by K features that each take one of n discrete values (Figure 2.1C and see *Definition of the stimuli* in *Methods*). As a simple

example, one feature could be shape, and two values for shape could be square or triangle; a second feature could be color, and two values could be red or blue. In all, there are n^K possible stimuli. So, there are four stimuli in our example. For each stimulus, the likelihood of making an error is the same (see *Definition of the stimuli* in *Methods*). In that way, the results we describe here do not depend on the distribution of stimuli. As an example, even if red squares were far more likely to occur than any of the other three stimuli, the error rate would be the same if they were all equally probable. However, in the case where red squares are far more likely than the other stimuli, it would be possible to design a code that dedicates more resources to discriminating red squares than to discriminating the less probable stimuli that would potentially have superior performance to the codes that we study here. We discuss this possibility further in the Discussion and *Linear transform (β)* in *Methods*. Finally, while we focus on discrete features, our core results are the same with continuous features (see *Error-reduction by mixed selectivity in the continuous case* in *Supplement*).

To understand how mixed selectivity affects code reliability and efficiency, we compare the performance of codes with different levels of conjunctive stimulus feature mixing, following the definition of nonlinear mixed selectivity used previously in the literature[62, 63]. We refer to these different levels of mixing as the order of the neural code. In particular, neurons in a code of order O respond to one particular combination of O feature values and do not respond otherwise (Figure 2.1D and see *Definition of the codes* in *Methods*), and a code has a neuron that responds to each possible combination of O different stimulus feature values (see *Code example* in *Methods* for more details). In our example, an order one ($O = 1$) code would have neurons that respond to each shape regardless of color and each color regardless of shape while an order two code ($O = 2$) would have neurons that respond to each combination of shape and color – for instance, one neuron would respond only to red squares, another only to blue squares, and so on. This example can map onto the two features used in the illustration

in Figure 2.1D, E. From this construction, each stimulus will have a unique response pattern across the population of neurons, but the population size will vary across code order. In general, higher-order codes will have larger population sizes, but less activity on average per neuron in the population.

With this formalization, we derive closed-form expressions for the population size (D_O), representation energy (P_O), and minimum distance (Δ_O) of our codes. These expressions are each functions of the number of features K, the number of values each of those features can take on n, and the order of the code O (Figure 2.1F). The population size for a code of order O, with K features that each take on n values is given by

$$D_O = \binom{K}{O} n^O$$

which can be viewed in terms of $\binom{K}{O}$ subpopulations that each encode all possible value combinations of O features, n^O. Following from this intuition, the representation energy for a code of order O is given by

$$P_O = \binom{K}{O}$$

That is, there is one neuron active in each of the $\binom{K}{O}$ subpopulations described above (see *Representation energy (P_O) of the codes* in *Methods* for more details). Finally, the minimum distance between responses in one of our codes is the distance between the responses to pairs of stimuli that differ only by one feature (though, in $O = K$ codes, pairs of stimuli that differ by more than one feature are also separated by the minimum distance, see Eq. 2.2 and *Code neighbors* in *Supplement*). Intuitively, this distance is related to the number of neurons

active for one stimulus, but not the other. The expression for minimum distance is

$$\Delta_O = \left[2\binom{K-1}{O-1}\right]^{\frac{1}{2}}$$

Here, $\binom{K-1}{O-1}$ gives the number of subpopulations that have different activity when one feature is changed, and the rest of the expression converts that number to the distance between the two representations. We go into more detail on each of these expressions in *Code properties* in *Methods*.

Using these expressions, we show that the ratio between squared minimum distance Δ_O and representation energy P_O is strictly increasing with order for all choices of K and n (see *Minimum distance-representation energy ratio* in *Methods* and Figure 2.1G, left):

$$\begin{aligned}\frac{\Delta_O^2}{P_O} &= \frac{2\binom{K-1}{O-1}}{\binom{K}{O}} \\ &= 2\frac{O}{K}\end{aligned} \tag{2.1}$$

This shows that, given the same amount of representation energy, codes with more mixing produce stimulus representations with strictly larger minimum separation in the response space (Figure 2.1G, left). Further, higher order codes also have a strictly lower amount of representation energy per neuron in the population (Figure 2.1G, right).

This increased separation between response patterns with increased code order results from the increased effective dimensionality of the response space of those codes. By effective dimensionality, we mean the smallest number of basis vectors (i.e., dimensions) necessary for a faithful linear reconstruction of the response space – this is equivalent to the number of non-zero eigenvalues in principal components analysis[125]. The effective dimensionality is always less than or equal to the population size of our codes. Intuitively, the higher

the effective dimension of the response space, the more response patterns can be arranged within it at a particular distance from each other given the same amount of representation energy. So, codes with higher effective dimensionality will usually have higher minimum distance. To illustrate this, we consider the $O = K$ case: Here, each stimulus is projected onto its own dimension in the response space. As a result, each stimulus representation will be at the maximum possible distance from all other stimulus representations, assuming only positive responses. If there were fewer effective dimensions than stimuli, it would no longer be possible to place all of the representations at this maximal distance from each other – so, minimum distance is necessarily decreased. Additionally, in the $O = K$ case, where the response dimension for each stimulus corresponds to a single neuron in the population, this leads to a hyper-sparse representation of the stimuli, where only one neuron fires for each stimulus. However, the same distance between stimulus representations is achieved for any rotation of the response dimensions relative to the neural population. This kind of rotation can be used to produce neural representations that match the heterogeneity of responses of real neurons, in which firing rates are both increased and decreased from the mean response (for an example, see Figure 2.4C). In particular, instead of hyper-sparse stimulus representations with a single active neuron each, the code can be rotated by the linear transform such that, in an extreme example, each stimulus is represented by a random Gaussian vector, in which almost all of the neurons in the population modulate their firing, and can therefore be considered active, in response to each stimulus (see *Linear transform (β)* in *Methods* for further discussion). Thus, it is not the sparsity per se that implies these effects, but the higher dimensionality of more, relative to less, mixed codes. While we believe that this distinction is conceptually important, the neural circuit implementation of codes with these rotations is likely to be more involved than for those without, and could have consequences for the efficiency and biological feasibility of these rotated codes.

In practice, the rotation of the response space relative to the neural population can be

achieved by the application of a linear transform to the response patterns of our codes (Figure 2.1A, β). In addition to altering the sparsity of neural responses for our codes via rotation, the linear transform can be used to rescale the representation energy used by each code without rotation. In the rest of the text, we only use the linear transform step of encoding to equate the representation energy of codes with different levels of mixing. The linear transform can also be used to expand the population size used by a code – and to exchange few neurons with high individual signal-to-noise ratios (SNRs) for many neurons with lower SNRs and redundant or partially redundant feature tuning. For instance, in our framework, one neuron firing ten spikes in response to a particular stimulus has the same representation energy and distances between stimulus representation as a code that replaces that neuron with two neurons that each fire approximately seven spikes for the same stimulus (due to our squared distance metric for representation energy; our core result holds for a sum of spikes metric as well, see Figure 2.5 and *Sum of spikes representation energy* in *Supplement*). As experimentally observed neural populations are often composed of neurons with heterogeneous firing rates and SNRs as well as partially redundant feature tuning (as exemplified below, in Figure 2.4), the linear transform can be used to make our codes exhibit activity that better matches the activity of real neural data.

Importantly, for independent and identically distributed noise that is applied to each neuron after the linear transform (as primarily studied here, Figure 2.1A, but see *Alternate noise models* in *Supplement*), the change in representation energy produced by the linear transform affects code performance, while the other manipulations discussed above do not. This is because the linear transforms used here (*Linear transform (β)* in *Methods*) cannot increase or decrease the effective dimensionality of the codes and do not change the underlying relative geometry of the stimulus response patterns to each other except by a uniform scaling. Instead, it is the nonlinear, conjunctive encoding of mixed codes that increases their effective dimensionality, and which produces their greater separation of stimuli in the response space

given the same amount of representation energy as pure codes. Due to this, we only use the linear transform to rescale representation energy in the following results.

As a result of their increased separation, mixed codes provide a benefit to decoding for many different noise distributions and decoders (including linear and maximum likelihood decoders), and indicates that mixed codes are likely to produce more reliable and efficient representations than pure codes in a wide variety of conditions. However, to directly quantify transmission reliability (i.e., the probability of a decoding error), we must include the details of both the noise and the decoder (see Figure 2.1A and *Full channel details* in *Methods*).

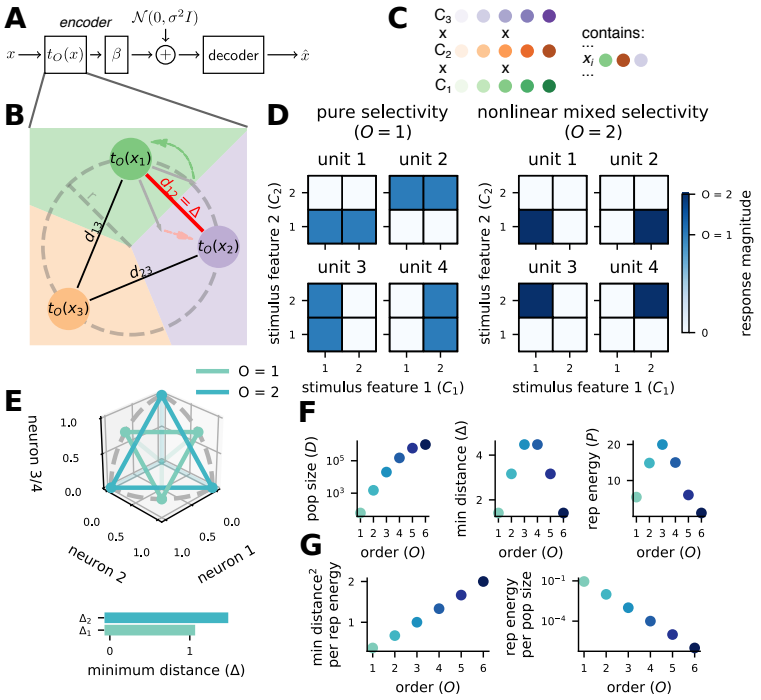

Figure 2.1: Mixed codes produce more discriminable stimulus representations. **A** The noisy channel model. A stimulus x is encoded by encoding function $t_O(x)$ of order O. Next, the linear transform β is applied before independent Gaussian-distributed noise with variance σ^2 is added to the representation. Finally, a decoder produces an estimate $\hat{x}$ of the original stimulus. **B** We analyze the encoding function with respect to three important code properties. The minimum distance $\Delta = d_{12}$ is the smallest distance between any pair of encoded stimuli (codewords), can be used to approximate the probability of decoding error. Representation energy $P = r^2$ is the square of the radius of the circle that all of the codewords lie on. Dimension is the number of neurons required to implement the code, here $D = 2$. **C** Stimuli are described by K features C_i which each take on $|C_i| = n$ values. All possible combinations of feature values exist, so there are n^K unique stimuli. **D** In pure selectivity (left), units in the code, or neurons, respond to a particular value of one feature and are invariant to changes in other features. In nonlinear mixed selectivity (right), neurons respond to particular combinations of feature values, and the number of feature values in those combinations is defined as the order O of the code (here, $O = 2$). **E** The same $O = 1$ and $O = 2$ code as in **D**. (top) The colored points are the response patterns in 3D response space for three of the four neurons in each code. The dashed grey line is the radius of the unit circle centered on the origin for each plane. For ease of visualization, the vertical dimension in the plot represents both the third and fourth neurons in the population to show three representations from the $O = 1$ code, this does not change the minimum distance. (bottom) The response patterns for the $O = 2$ mixed code have greater minimum distance than those for the $O = 1$ pure code. **F** We derive closed-form expressions for each code metric, and plots of each metric are shown for codes of order 1 to 6 with $K = 6$ and $n = 10$. **G** Mixed codes produce a higher minimum distance per unit representation energy (left) and have a smaller amount of representation energy per neuron than pure codes.

Mixed codes make fewer errors than pure codes

To directly estimate the probability of decoding error, or error rate, for each of our codes, we expand our analysis from the encoding function (Figure 2.1) to the channel as a whole. We choose the noise to be additive, independent, and Gaussian (though we also consider two kinds of Poisson-distributed noise, which give similar results, see Figure 2.6 and *Alternate noise models* in *Supplement*). Finally, for decoding, we use a maximum likelihood decoder (MLD; and see *Full channel details* in *Methods*). Given these noise and decoder assumptions, we can estimate the error rate by decomposing the probability that we make an error into the sum of the probabilities of only the most likely errors (that is, errors to stimuli at minimum distance; see *Estimating the error rate* in *Methods*). To proceed with our estimate, we first need to know how many stimulus representations are at minimum distance from a given stimulus for each code, which we refer to as neighbors at minimum distance or nearest neighbors. For stimuli with K features that each take on n values encoded by a code that conjunctively mixes every combination of O features, this is given by,

$$N_{\Delta}(O) = \begin{cases} K(n-1) & O < K \\ n^K - 1 & O = K \end{cases} \tag{2.2}$$

To obtain this expression, we show that the distance between stimulus pairs that differ in only one feature is strictly smaller than between those that differ by two features for all codes with $O < K$. That is, only pairs of stimuli that differ by a single feature will be at minimum distance from each other, all other pairs of stimuli will be separated by a larger distance. Since there are $K(n-1)$ stimuli that differ from each stimulus by one feature, that is the number of neighbors each stimulus has at minimum distance for all codes $O < K$. For the $O = K$ code, since $\binom{K}{K} = 1$, the code can be viewed as having a single subpopulation. That subpopulation will have different activity for every other stimulus, no matter how many

features those other stimuli differ by. Thus, all $n^K - 1$ other stimuli are at minimum distance from a particular stimulus. We derive these expressions more formally in *Code neighbors* in *Supplement*.

Now, given the number of neighbors at minimum distance, the minimum distance itself, and the assumption of additive Gaussian noise, our estimate of the error rate (PE) takes the following form:

$$\begin{aligned} PE &\approx N_\Delta(O)\, Q\left(\frac{\Delta_O}{2\sqrt{P_O}}\mathrm{SNR}\right) \\ &= N_\Delta(O)\, Q\left(\frac{\mathrm{SNR}}{\sqrt{2K/O}}\right) \end{aligned} \tag{2.3}$$

where $Q(y)$ is the complementary cumulative distribution function of the standard normal distribution at y, $\mathrm{SNR} = \sqrt{V/\sigma^2}$ is the population signal-to-noise ratio (see *Linear transform (β)* in *Methods*), and $N_\Delta(O)$ is the number of neighbors at minimum distance for the code of order O, defined above. This estimate reveals that, for the same SNR, increasing the order of our codes will strictly decrease the probability of a decoding error for codes with order $O < K$. In *Estimating the error rate* in *Methods*, we use a more detailed estimate to show that the $O = K$ code will have an even lower error rate than any code with $O < K$. That is, the $O = K$ code will always be the most efficient and robust, given this method of accounting for representation energy and metabolic cost.

To verify that our estimate of the error rate is accurate, we numerically simulate codes of all possible orders over a wide range of SNRs for particular choices of K and n using the same channel as in our analysis. Our simulations show that higher-order codes outperform lower-order codes across all SNRs at which the codes are not saturated at chance or at zero error (Figure 2.2A). We also show that the estimate closely follows performance for large SNRs (Figure 2.2A insets). Using this estimate, we compare the error rate of different codes

at fixed, high SNR (Figure 2.2B) and show that pure codes make several orders of magnitude more errors than the mixed code with the optimal order. This also illustrates that, for larger K, the full-order mixed code ($O = K$) and close to full-order codes (O close to K) have similar performance (Figure 2.2B). Due to their smaller population sizes, codes with less than the full amount of mixing (order near K) may be desirable in some cases. We make this intuition explicit by accounting for the metabolic cost of neural population size in the following section. In all conditions we simulated (in agreement with our estimate), the fully mixed ($O = K$) code had the lowest error rate at a given SNR, though other highly mixed codes (O near K) reached nearly equivalent error rates with larger numbers of features (K; Figure 2.2A, bottom). Thus, in these conditions, mixed codes provide a significant benefit to coding reliability independent of particular parameter choices.

For smaller choices of K and n, we were able to empirically evaluate how decoding error compares to the rate-distortion bound[61]. In this context, the rate-distortion bound is an absolute lower bound on the probability of making a decoding error given a particular information rate through the channel (i.e., the mutual information $I(X; \hat{X})$; see *The rate-distortion bound and mutual information calculation* in *Supplement* and Figure 2.8A). We first show that higher-order codes generate a higher information rate than lower-order codes at most SNRs (Figure 2.8B, inset) – that is, they more efficiently transform the input into stimulus information. Next, we show that the full-order code ($O = K = 3$) fully saturates the rate-distortion bound (Figure 2.8B). Thus, for a given amount of stimulus information, full-order codes produce as few errors as would be possible for any code[61]. While the $O = K - 1 = 2$ code comes close to this bound as well, the pure code does not.

In the above, we have focused on the case with noise applied only to the neural responses in our codes; however, we also show how our codes are affected by two kinds of noise applied to the input stimulus representation (see *Alternate noise models* in *Supplement*). While input

noise is often impossible to correct completely[126], we show that input noise that perturbs the input to nearby stimulus values does not affect codes of different orders differently. That is, even though such noise is uncorrectable by our codes, it leads to the same error rate for codes with all levels of mixing (Figure 2.7A, B). Next, we show that input noise that is not confined to perturbations to nearby stimulus values does differentially affect codes of different orders, and can lead to higher error rates in some more mixed codes (Figure 2.7C). However, we also show that mixed codes still outperform pure codes in many cases due to their greater robustness to output noise (Figure 2.7D and Figure 2.2).

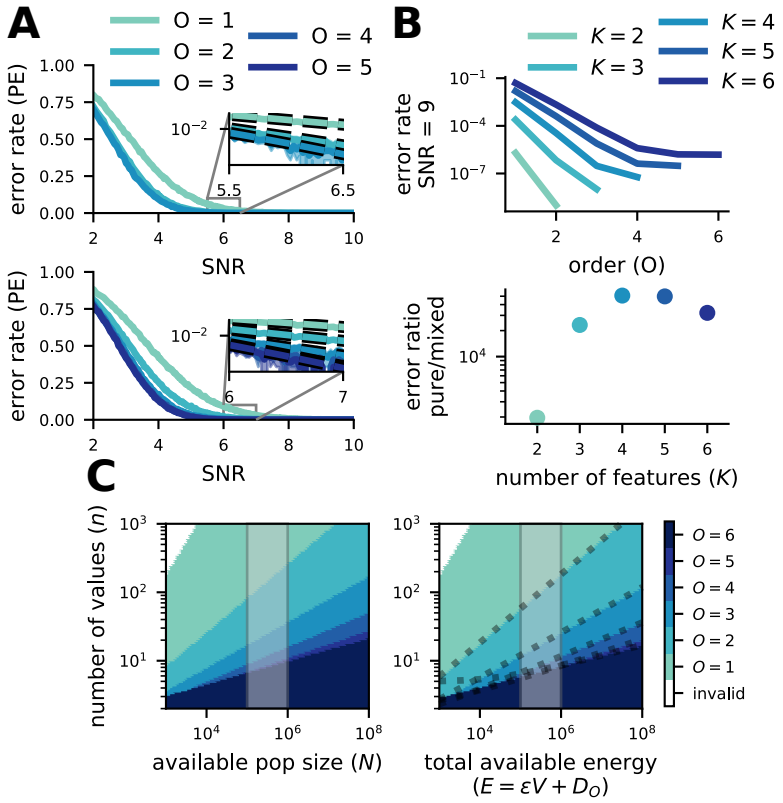

Figure 2.2: Mixed codes make fewer errors than pure codes. **A** (top) Simulation of codes with $O = 1, 2, 3$ for $K = 3$ and $n = 5$. (inset) For high SNR, code performance is well-approximated by our estimate of error rate. (bottom) Same as above, except with $K = 5$ and $n = 3$. **B** (top) The estimated error rate at a fixed, high SNR (SNR $= 9$) for codes of every order given a variety of different K (all with $n = 5$). Error probability decreases with code order for all codes except, in some cases, the $O = K$ code. (bottom) The number of errors made by the pure code for every error made by the optimal mixed code at SNR $= 9$ (as above). In all cases, pure codes make several orders of magnitude more errors than the optimal mixed code. **C** (left) Given a pool of neurons with fixed size, the color corresponding to the code producing the highest minimum distance is shown in the heat map. The shaded area delineates the order of magnitude of the number of neurons believed to be contained in $1\,\text{mm}^3$ of mouse cortex. (right) The same as on the left, but instead of a pool of neurons of fixed size, each code is given a fixed total amount of energy. The energy is allocated to both passive maintenance of a neural population (with size equal to the population size of the code) and representation energy (increasing SNR). The shaded area is the same as on the left. The dashed lines are plots of our analytical solution for the transition point between the O and $O + 1$-order code (see *Total energy* in *Methods*).

2.3.2 Mixed codes provide benefits despite requiring more neurons

Our analysis so far has focused on the metabolic cost of neuronal spiking. A single spike is thought to be the largest individual metabolic cost in the brain[127]. For a fixed population size N, from Eq. 2.1, we know that the code with the highest order O such that $D_O \leq N$ will provide the largest minimum distance, given a fixed amount of spiking activity (Figure 2.2C, left). For a wide range of stimulus set sizes, mixed codes have population sizes less than or equal to an order-of-magnitude estimate of the neuron count in $1\,\text{mm}^3$ of mouse cortex[128] (Figure 2.2C, left, shaded region). Thus, the benefits of mixed codes are practically achievable in the brain.

However, the passive maintenance of large neural populations also has a metabolic cost, due to the turnover of ion channels and other cell-level processes[127], which, for large populations of sparsely firing neurons, could be as large if not larger than the metabolic cost associated with spiking. To account for this cost, we adapt the formalization from [129] to relate representation energy (i.e., spiking) to the metabolic cost of population size. We refer to the sum of these costs as the total energy E of a code (see *Total energy* in *Methods*). Codes with small population sizes will be able to allocate more of their total energy to representation energy, while codes with large population sizes will have less remaining total energy to allocate to representation energy. We do not constrain the maximum SNR that a single neuron in our codes can achieve (even though achievable SNR is limited in the brain[130]). Due to the exponential growth of population size with code order, this choice favors pure codes over mixed codes. In particular, without limiting single neuron SNR, pure codes can allocate the majority of their total energy to the activity of relatively few neurons due to their small population size; if we were to limit single neuron SNR, then pure codes would also have to grow their population size with increased total energy, which would decrease the fraction of that energy used for representation energy. Thus, this analysis serves as a

particularly stringent test of the reliability and efficiency of mixed codes.

Mixed codes yield higher minimum distance under the total energy constraint for a wide range of stimulus set sizes and total energy (Figure 2.2C, right), including order-of-magnitude estimates of the total energy available to $1\,\mathrm{mm}^3$ of mouse cortex (Figure 2.2C, right shaded region). Further, our analysis reveals that for any total energy $E \geq n^2K^2$ (see Eq. 2.12) a mixed code ($O > 1$) will provide better performance than the pure code ($O = 1$). These results also make an important prediction that can be tested experimentally: the order of neuronal RFs should decrease as the fidelity required of the representation increases (i.e., as n increases). There already exists indirect experimental support for this prediction. In the visual system, single neurons in primary visual cortex have RFs thought to represent relatively small combinations (small O) of low-level stimulus features such as spatial frequency and orientation[51, 64] (but see [131]), while single neurons in the prefrontal cortex are thought to have responses that depend on larger combinations (high O) of abstract, often categorical (and therefore low n), stimulus features along with behavioral context[62, 63]. However, this pattern has not been rigorously tested, as these regions are rarely recorded in the same tasks and the tasks chosen for each area often follow the form of the prediction – that is, requiring high fidelity (n) for investigations of primary sensory areas and low fidelity (n) for investigations of prefrontal areas.

2.3.3 Mixed codes provide reliable coding in sensory systems

So far, we have focused on the probability of decoding error, which is most applicable to features that represent categorical differences without defined distances from each other (e.g., mistaking a hat for a sock is not clearly less accurate than mistaking a hat for a glove). However, in sensory systems, the features often do have a relational structure and stimuli that are nearby to each other in feature space are also perceptually or semantically similar (e.g., mistaking a 90° orientation for a 180° orientation is clearly less accurate than

mistaking 90° for 100°). In the context of sensory information, minimizing the frequency of errors becomes less important than ensuring that the average distance of an estimate from the original stimulus is low. This is because perceptually similar errors are likely more useful for guiding behavior than a random error, even if the latter occurs less frequently. This difference in priority is encapsulated in the contrast between error rate (Figure 2.3B) and the mean squared-error distortion (MSE; Figure 2.3C), which is equivalent to the average squared-distance of the estimated stimulus from the original stimulus. In our framework, full-order mixed codes have the highest minimum distance (Eq. 2.1), but all stimuli are nearest neighbors to all other stimuli (Eq. 2.2) which causes all errors to be random with respect to the original stimulus. Using MSE instead of error rate, we show that lower-order mixed and pure codes outperform full-order mixed codes at low total energy (Figure 2.3C). However, increased total energy causes a faster decay in error rates for full-order codes than lower-order codes (Eq. 2.3), so full-order codes outperform pure codes even under MSE at high total energy (Figure 2.3B).

Further, experimental investigation of sensory brain regions often reveal neurons with response fields (RFs) that include multiple (sometimes many) perceptually similar stimuli. To investigate how these response fields affect the error rate and MSE of our codes, we generalize our formalization to include RFs (Figure 2.3A), that can take on different widths, written as σ_{rf}. Here, instead of responding to a particular combination of O feature values, neurons in a code of order O will respond when the value of each of O features fall within a particular interval of values, with length σ_{rf}. Thus, each neuron in an order O code will respond to a contiguous region of σ_{rf}^{O} stimuli, as illustrated for an $O = 2$ code with $\sigma_{\text{rf}} = 2$ in Figure 2.3A. This generalization introduces a new dependence of both representation energy and population size on RF width. For representation energy, the generalization is simple:

representation energy grows linearly with RF size,

$$P_O = \binom{K}{O}\sigma_{\text{rf}}$$

because σ_{rf} neurons in each population need to be active to unambiguously identify a single size O feature-value combination (Figure 2.10B). For population size, it has previously been shown that increasing RF size can vastly decrease required population size[132, 133]. Here, we find that

$$D_O = \binom{K}{O}\sigma_{\text{rf}}\left(\frac{n}{\sigma_{\text{rf}}}+1\right)^O$$

for $O > 1$, while for $O = 1$ the population size does not change with σ_{rf}. From this expression, we see that the population size for $O > 1$ codes decreases approximately as $1/\sigma_{\text{rf}}^{O-1}$ with RF size (Figure 2.10A). Minimum distance does not depend on RF size. In terms of total energy, these dual dependencies on RF size largely cancel each other out, constructing a code with the RF size chosen to minimize total energy consumption does not typically lead to a change in the code order that maximizes minimum distance (Figure 2.10E, G).

Instead, the principal benefit produced by increasing RF size is the reduction of MSE for all codes that results from making errors to nearby stimuli in stimulus space more likely. This is because stimulus decoding now depends on the simultaneous activity of neurons with overlapping RFs, and the most likely errors are now those in which one of that group of neurons is confused for a different neuron that also has an overlapping RF with the rest of the group. Thus, for the full-order code, increasing RF size significantly reduces the randomness of errors and allows the brain to take advantage of their increased minimum distance in the context of sensory systems (Figure 2.3C). Thus, this work provides a unified framework for understanding the purpose and benefits of large RFs in arbitrary feature spaces, which are often observed in cortex[134]. In particular, increasing RF size decreases

the MSE for all codes while increasing the error rate (Figure 2.3D and see *Additional results on response fields* in *Supplement*). The increase in error rate results from the increase in representation energy required for the code without an associated increase in minimum distance; the decrease in MSE results from the fact that the additional representation energy provides information about the stimulus space, which causes errors for larger RF codes to be closer to the original stimulus (Figure 2.3E). Intuitively, for the the $O = K$ case, increasing RF size makes stimulus representations non-orthogonal (this can also be viewed as making them less sparse in some conditions), which means that they are no longer positioned at the maximum possible distance from each other (so, the error rate is increased); but, it also means that nearby stimuli now have more similar representations, which makes them more likely errors and leads to the reduction in MSE, correcting the undesirable feature of randomly distributed errors for full-order codes (see *Additional results on response fields* in *Supplement*).

We also show increased noise robustness from mixed codes in simulations of a code for continuous stimuli under MSE, using continuous RFs (Figure 2.11A). Thus, mixed codes are an effective strategy for reliable and efficient coding not just for decision-making systems, but also in sensory systems – which is consistent with their widespread observation in sensory brain regions[64–69].

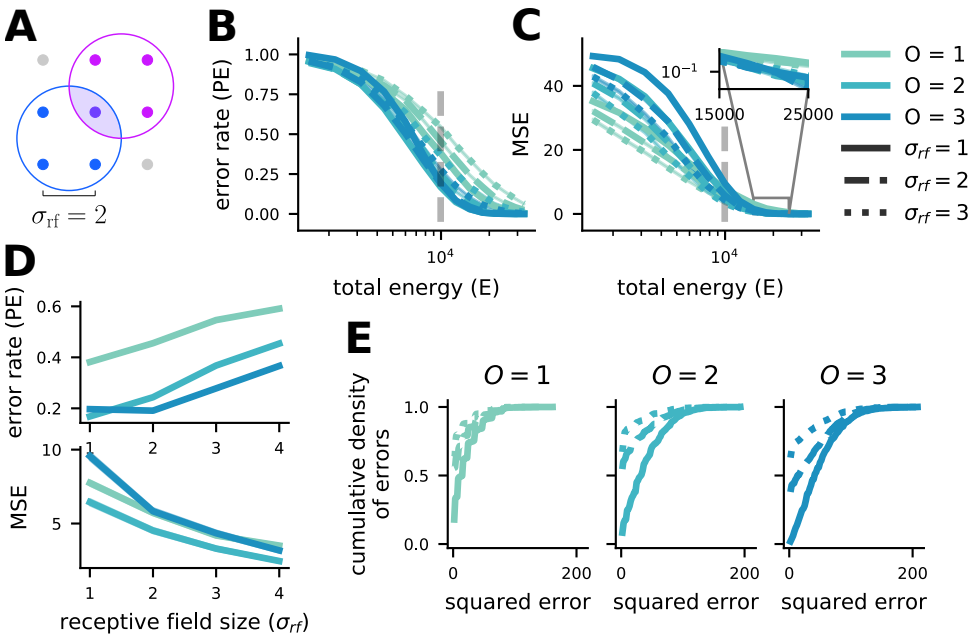

Figure 2.3: Mixed codes can be more reliable than pure codes for both PE and MSE, but different RF sizes are appropriate for each. **A** An illustration of our RF formalization. With $K = 2$ and $n = 3$, two example RFs of size $\sigma_{\mathrm{rf}} = 2$ are shown. Simultaneous activity from both neurons uniquely specifies the center stimulus point. **B** Simulated PE of codes of all orders for $K = 3$ and $n = 10$ with $\sigma_{\mathrm{rf}} = 1, 2, 3$ (legend as in **C**). Note that total energy is plotted on the x-axis, rather than SNR as in Figure 2.2. Mixed codes outperform the pure code over many (but not all) total energies. **C** The same as **B** but for MSE rather than PE. Mixed codes perform worse than pure codes for low total energy, but perform better as total energy increases. **D** PE increases (top) and MSE decreases (bottom) as σ_{rf} increases for the codes in **B** and **C** taken at the total energy denoted by the dashed grey line. **E** Cumulative distribution functions for the squared errors made by the codes given in **B** and **C** at the grey dashed line. MSE is decreased by increasing σ_{rf} despite the increase in PE because the errors that are made become smaller in magnitude and this outweighs their becoming more numerous. This effect is largest for the $O = K = 3$ code.

2.3.4 Experimental evidence that mixed codes support reliable decoding

Several previous theoretical studies of mixed selectivity have focused on the fact that it enables flexible linear decoding, and there is experimental evidence that the dimensionality expansion provided by mixed selectivity is linked to performance of complex cognitive behaviors[62, 63]. Here, we have shown that mixed codes also provide more general benefits for reliable and efficient information representation in the brain, independent of a particular task and without assuming linear decoding. Thus, our work predicts that mixed codes will be used widely in the brain, instead of being used only for features relevant to particular complex tasks.

To understand whether the brain exploits mixed codes for their general reliability and efficiency rather than only for their ability to enable flexible computation, we test whether the brain implements mixed selectivity when it would not enable the implementation of any behaviorally relevant linear decoders. To do so, we analyze data from a previously published experiment[135] that probed how two behaviorally and semantically independent features are encoded simultaneously by neurons in the lateral intraparietal area (LIP). In the experiment, monkeys performed a delayed match-to-category task in which they were required to categorize a sample visual motion stimulus (Figure 2.4A), and then remember the sample stimulus category to compare with the category of a test stimulus presented after a delay period (Figure 2.4B, top). In addition to the categorization and working memory demands of the task, the animals were also (on some trials) required to make a saccadic eye movement either toward or away from the neuron's RF during the task's delay period (Figure 2.4B, bottom, and see *Experimental details and task description* in *Methods*). Because LIP activity is known to encode information related to categorical decisions and saccades, this experiment characterized the relationship between the representation of these two features at the single neuron and population level. Despite the saccade being irrelevant to the monkey's categori-

cal decision in this task, LIP activity demonstrated both pure ($O = 1$, 40/61 neurons were tuned for at least one pure term) and mixed category and saccade tuning ($O = 2$, 31/61 for at least one mixed term; Figure 2.4D, E). This pattern of mixed and pure tuning is consistent with a composite code including RFs of multiple orders. Such codes have performance that falls between codes of either the lowest or highest included order alone, but their heterogeneity may provide other benefits. In particular, a composite $O = K$ and $O = K - 1$ code would have a minimum distance per representation energy ratio between that of each code alone, but would have the same number of nearest neighbors as the $O = K - 1$ code (that is, $K(n - 1)$ rather than $n^K - 1$ nearest neighbors). Thus, in some cases, this composite code may provide lower MSE distortion than either the $O = K$ or $O = K - 1$ code alone.

Crucially, mixed codes also provide benefits when decoding only one of the two features at a time with a maximum likelihood decoder (Figure 2.4F). This results from the increased separation between all response patterns produced by mixed codes. However, the same tradeoff that we demonstrated between fully-mixed ($O = K$) and less mixed ($O < K$) codes for stimulus identity decoding applies to single feature decoding as well. That is, the number of likely errors that result in the decoding of a different feature value is larger for fully-mixed codes than for less-mixed and pure codes, but each of those errors is less likely. To illustrate this, we consider the current case, with $K = 2$ and $n = 2$. Here, for the $O = 1$ code, there is one possible kind of noise perturbation that would lead to an error: one that brings the component corresponding to the correct value of the target feature lower than the component corresponding to the incorrect value of the target feature. Thus, in general, there are $n - 1$ ways to make errors for the $O = 1$ code, and for all the $O < K$ codes, following from statement 3. For the $O = 2$ code, there are two kinds of noise perturbations that would cause an error, since there are two components that correspond to the incorrect value of the target feature, rather than just one. In general, there are $(n - 1)n^{K-1}$ ways to make errors for the $O = K$ code. Thus, we can write an estimate for the single feature decoding error

rate that is analogous to Eq. 2.3:

$$PE_{\mathrm{f}} \approx N_{\Delta,\mathrm{f}}(O)\, Q\left(\frac{\mathrm{SNR}}{\sqrt{2K/O}}\right)$$

where PE_{f} is the probability of making a single feature decoding error and $N_{\Delta,\mathrm{f}}(O)$ is the number of neighbors with the incorrect feature value that a code of order O has at minimum distance. This is written as,

$$N_{\Delta,\mathrm{f}}(O) = \begin{cases} n-1 & O < K \\ (n-1)n^{K-1} & O = K \end{cases}$$

following from the discussion above. As a result, the utility of mixed codes for single feature decoding is similar to their utility for stimulus identity decoding: For smaller values of K, the full-order ($O = K$) code will provide the best performance due to its maximal separation of stimulus representations; as K grows larger, codes with close to full mixing (O near K) will begin to all provide equivalent performance.

With two behaviorally and semantically independent features, the brain still implements a mixed code even though it does not enable the implementation of any behaviorally useful linear decoders. The mixed code does, however, improve the reliability and efficiency of the encoding for both features separately and when combined, suggesting that the brain may explicitly utilize mixed codes for that purpose. Further, contemporaneous work has demonstrated that the bat is likely to exploit the reliability benefits of mixed selectivity for the coding of two-dimensional continuous head-direction information – as well as described reliability benefits of full-order mixed codes for continuous stimuli[65] (and see *Error-reduction by mixed selectivity in the continuous case* in *Supplement*).

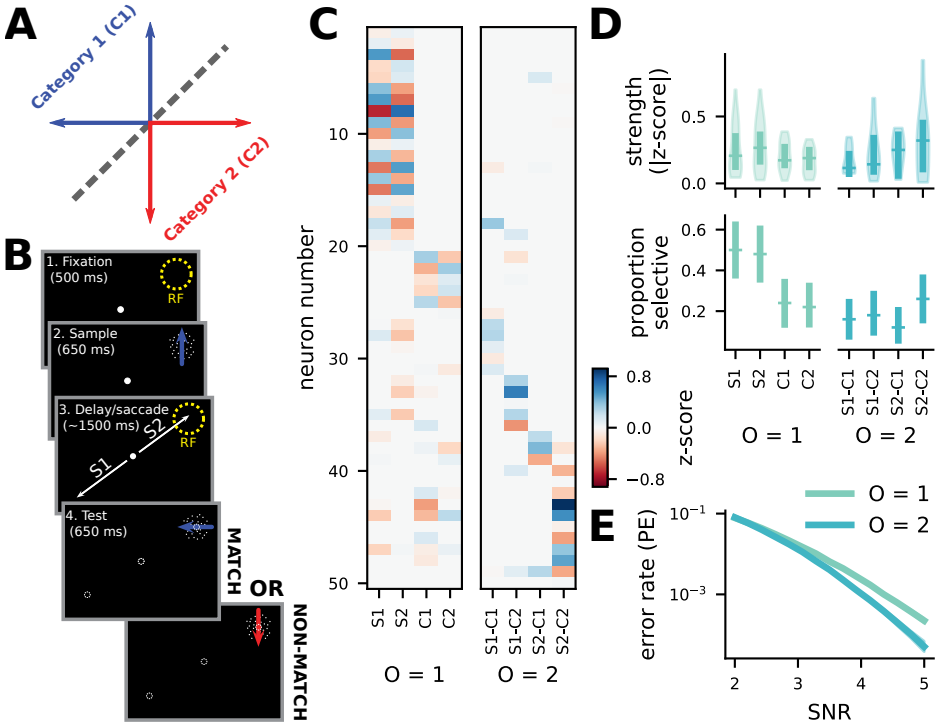

Figure 2.4: Mixed codes support reliable decoding in the brain, not only flexible computation. **A** The learned, arbitrary category boundary on motion direction used in the saccade DMC task. **B** A schematic of the saccade DMC task. **C** A heatmap of the z-scored magnitude of the coefficients for each term in the linear model. It is sorted by largest magnitude term, from left to right. The linear models were fit using the LASSO method and terms were tested for significance using a permutation test ($p < .05$), only neurons with at least one significant term were included in this and the following plots (50/61 neurons). In addition, 10 out of 71 total recorded neurons were excluded due to having less than 15 recorded trials for at least one condition. **D** (top) The average strength of significant tuning for each term across the neural population, $O = 1$ tuning is on the left, and $O = 2$ tuning is on the right. (bottom) The proportion of neurons in the population that have pure selectivity (left) for the two saccade targets and two categories of motion and nonlinear mixed selectivity (right) for each of the four saccade target and category combinations. Error bars are bootstrapped 95 % confidence intervals. **E** Single-feature decoding performance for a code chosen to mirror the conditions of the task, with $K = 2$ and $n = 2$. Mixing features together is advantageous even when decoding those features separately.

2.4 Discussion

We have shown that mixed selectivity is an effective and general strategy for reliable and efficient communication. Further, we have demonstrated that, rather than pure ($O = 1$) or fully-mixed ($O = K$) codes always providing the most reliable encoding, the optimal code order tends to lie between these two extremes ($1 < O < K$) depending on the number of stimulus features, the required fidelity of those features, and the number of neurons or total metabolic energy available to encode the information (Figure 2.2D). This set of intermediately mixed codes has not previously been analyzed in this context, despite likely being the dominant form of mixed selectivity that exists in the brain. Intermediately mixed codes may also have an important additional benefit. The representations produced by the full-order mixed code ($O = K$) in our framework may be difficult to learn and to generalize from[74, 136], due to the fact that each response pattern is the same distance from all other response patterns. Intermediately mixed codes ($1 \leq O < K$) ameliorate this by placing response patterns that are nearby in stimulus space nearby to each other in response space as well. For $O = K$ codes, we have also shown that nearby stimuli can be placed nearby to each other in response space by increasing RF size, though this increases the representation energy required by the code. This means that intermediately mixed codes and full-order codes with larger RFs carry information not just about the encoded stimulus, but also about which stimuli are nearby to the encoded stimulus, while full-order codes with $\sigma_{\text{rf}} = 1$ carry information only about the encoded stimulus. This information about which stimuli are nearby is likely to be crucial for behavioral performance and learning[73]; however, carrying this extra (from the perspective of stimulus decoding) information often increases the error rate. Lastly, we have shown experimental evidence that the brain implements mixed codes even when they do not facilitate behaviorally relevant linear decoding, but do improve the reliability and efficiency of encoding.

This work differs substantially from most previous work on optimal RFs in four principal ways. First, the dependence of code reliability on RF order, or dimensionality, has not been comprehensively described. While previous work has shown that optimal RF width depends on the dimensionality of the stimulus[111, 137], stimulus dimensionality and RF dimensionality were assumed to be the same. Thus, the effect of changing RF dimensionality in conjunction with RF width was not explored, as it is here (see Figure 2.3 and Figure 2.11). We show that codes using RFs of intermediate dimension ($1 < O < K$) are most reliable in a wide variety of cases (Figure 2.2D), but these codes have not been previously studied outside of binary stimulus features. Second, we directly compute the probability and magnitude of errors for our codes rather than maximizing quantities such as Fisher information and mutual information. This reveals the performance of our codes in low SNR regimes and for different metrics of decoder performance (i.e., error rate and MSE). Third, by accounting for the metabolic cost of both the total spike rate as well as the minimum population size required to implement each of our codes while keeping coverage of the stimulus space constant, we disentangled performance decreases due to a lack of coverage of the stimulus space from those due to the properties of the encoding itself. Fourth, we have investigated differences in code performance across different orders for both discrete and continuous stimuli as well as both binary (error rate) and distance (MSE) error metrics. These different contexts have revealed several nuances, including that, for discrete stimuli, increasing RF size tends to increase error rate, but decrease MSE – highlighting the ways in which RF shape and size can influence which kinds of coding errors are likely for different coding strategies, which has not received extensive study in neuroscience. Thus, this work provides a novel perspective on multiple understudied neural coding problems.

This work also ties directly to existing work in the experimental and theoretical neuroscience literature. Most centrally, we link the previously described flexible linear decoding benefits of mixed selectivity to considerations of reliability and efficiency in neural codes. Exper-

imental work focusing on the utility of mixed selectivity for flexible linear decoding has already demonstrated the ubiquity of mixed codes in prefrontal cortex[63], as well as a putative link from mixed selectivity to behavior[62]. Previous theoretical work has shown that mixed codes with representation rescaling and population expansion can be constructed by the brain naturally in a variety of conditions, due to the nonlinearity of the neuronal input-output function. In particular, this work has demonstrated that random connectivity both in feedforward, binary-thresholded model neurons[120] and in recurrently connected neural network models[74] produces mixed codes for stimulus features. However, randomly constructed networks would require many more neurons than necessary to construct full codes of orders close to K. This concern is ameliorated by the learning rules that have been shown to be at work in cortex. Theoretical work that applies biologically plausible, unsupervised Hebbian-like plasticity rules to model networks similar to those in [74, 120] demonstrates that these rules can increase the prevalence of mixed selectivity to levels consistent with those observed experimentally in prefrontal cortex, which has been shown to have more diverse mixed selectivity than expected due to purely random connections[60].[138]. Thus, not only does this class of mixed codes provide two substantial and separate benefits to the brain (i.e., reliability and linear separability), they are also naturally produced by known neural phenomena – that is, they do not require fine tuning. However, mixed codes with a linear transform that rotates the stimulus representations with respect to the neural population and breaks the link between sparsity and code order, as discussed above and in *Linear transform (β)* in *Methods*, have a less clear neural circuit implementation. While these codes can replicate the heterogeneity in tuning observed in neural recordings, their implementation is likely to require either extensive nonlinear dendritic processing or recurrent interactions between neurons in the code.

Further, other work in theoretical neuroscience has explored how mixed selectivity can facilitate associative learning[120, 139] as well as negotiate a tradeoff between discrimination

and generalization[74]. However, the formalization used in these works differs substantially from the one we use here – and, while our results are broadly consistent with each other (i.e., that the nonlinear dimensionality expansion produced by mixed codes provides benefits to neural computation), these works differ from the current one in the precise level of mixing that they find as optimal. In particular, this previous work finds that lower levels of mixing are optimal, while we show that near-maximal mixing is optimal in our framework in many conditions. Our different conclusions arise from a key difference in our results: in our work, the effective dimension of our codes strictly grows with code order; in the other works, the effective dimension of the codes peaks at relatively low code orders[120, 139]. We believe that this difference arises primarily from two differences in our formalizations. First, we focused on codes with population sizes large enough to implement every combination of inputs at a particular level of mixing – in fact, population size is a function of code order in our framework. The other works, instead, use fixed population sizes across code orders and neurons were given selectivity for random subsets of input combinations. While this may be more biologically plausible in some contexts, previous theoretical work, as discussed above, has shown that biologically plausible plasticity rules can produce mixed selectivity that is significantly different from random, matching patterns observed in cortex[138]. Second, neurons in our codes respond to size O conjunctions of particular feature values, which produces a receptive field-like response structure; while, in the other works, neurons respond when the weighted sum of a subset of O stimulus feature values exceed a particular threshold, which can produce neurons with more heterogeneous responses. We believe that both of these differences together produce greater dimensionality expansion for large O in our framework as compared to the other works, and lead to our different conclusions. While our formalization is, in some ways, less mechanistic than these previous works, we believe that it does more clearly isolate the relative contributions of code order, population size, and representation energy to decoding performance.

Further, interrogation of the bat head-direction system has revealed a dynamic code that appears to shift between mixed and pure representations from moment to moment[65]. The bat head-direction system encodes both the azimuth and pitch of the animal[140], and thus could use either a pure ($O = 1$) or mixed ($O = 2$) code. Surprisingly, the brain appears to use both codes, and adjust the number of neurons with pure or mixed selectivity dynamically depending on the behavioral regime of the animal. In particular, when the animal is maneuvering on short timescales with high angular velocity (i.e., low SNR), the code is biased toward mixed selectivity; and when the animal is navigating over long distances with low angular velocity (i.e., high SNR), the code is biased toward pure selectivity[65]. The authors go on to show that, for a neural population of a size similar to that of the bat head direction system, this dynamic shift is the optimal strategy, as, while the mixed code provides lower decoding error on short timescales, the pure code can produce a finer-grained representation of the full head-direction space and lead to overall smaller errors on long timescales. This work indicates that both mixed and pure codes are decodable by the brain and indicates that the most reliable and efficient code is selected moment-to-moment as the timescale of decoding shifts. This also illustrates an important area for future research in our framework, as the representation energy available to a code is likely to be able to shift from moment-to-moment, while the population size of the code likely cannot. This may have consequences for the optimal code in more dynamic environments. More generally, mixed codes have been observed across diverse sensory and non-sensory systems[62–72, 141], indicating that their usefulness is not only due to enabling flexible linear decoding, but also due to their coding reliability and efficiency.

Our work also points to several areas for future research that will lead to a more detailed characterization of sensory feature representations across different brain regions. First, we have shown that the optimal code order decreases as the fidelity of the stimulus features (i.e., the number of values each feature takes on, n) increases. Thus, it will be important to

directly compare the fidelity and code order across different levels of the sensory-processing cortical hierarchy. Second, while the benefits of mixed relative to pure codes that we describe here do not rely on a particular stimulus distribution, it is possible that the performance of some codes could be improved by incorporating information about correlations between stimulus features. Intuitively, codes with an order at least the order of the feature correlation structure (e.g., $O \geq 2$ codes for pairwise feature correlations, $O \geq 3$ codes for triplet feature correlations, and so on) would be well-suited for this, as the individual SNR of single neurons in the code selective for particular feature combinations could be either increased or decreased if those feature combinations are more or less likely. As an example, if a particular combination of two feature values is more likely than other combinations of those same two features, then any code with $O \geq 2$ could statically adjust the individual SNR of neurons coding for that feature value combination to maximize performance; but, it would not be possible for an $O = 1$ code to make the same adjustment without assuming a network interaction effect that adjusts the SNR only when the neurons that code for each feature value independently are both active. However, further work is needed to determine whether this strategy improves performance for our codes when accounting for the difference in overall representation energy of the adjusted codes, as well as to determine how the adjustment could be learned in a biologically feasible way.

Overall, our work has shown that mixed selectivity is an effective and practical strategy for reliable coding in the brain. Guaranteeing this reliability, in the face of unreliable neurons, is likely to have fundamentally shaped the functional and even anatomical architecture of neural systems. Developing an understanding of the role of code order, or RF dimensionality, in reliable and efficient coding will give insight into this much broader problem.

Acknowledgments: We gratefully acknowledge Chris Rishel and Gang Huang for conducting the experiments which provided the neurophysiological data. We also thank Xaq

Pitkow, Jeff Beck, Nicolas Masse, Jared Salisbury, Krithika Mohan, and Yang Zhou for their comments on and useful discussion of earlier versions of this manuscript. This work was supported by NIH F31EY029155 (WJJ), NSF CAREER-1652617 (SEP), NIH R01EY019041(DJF), CRCNS NIH R01MH115555 (DJF), NSF NCS 1631571 (DJF), and a DOD Vannevar Bush Fellowship (DJF).

Author contributions: WJJ conceived of the project. SEP and DJF supervised the project development. WJJ created the model and performed the calculations, model simulations, and data analysis. DJF designed and supervised the experimental work. SEP supervised the theoretical work. WJJ, SEP, and DJF wrote the paper.

Competing interests: The authors declare no competing interests.

2.5 Methods

2.5.1 Definition of the stimuli

Our stimuli are defined as having K features, which each take on a single discrete value. Assuming that our stimuli are discrete simplifies our mathematical analysis, but also makes our analysis relevant to cognitive, categorical representations. In addition, simulations with continuous stimulus features have qualitatively replicated our core results (see *Error-reduction by mixed selectivity in the continuous case* in *Supplement*).

Here, a stimulus is represented by a vector of K discrete values. Each value corresponds to one of the K features of the stimuli. The nature of the value object does not matter, we only require that it is possible to decide whether two values corresponding to the same feature are equal. For a stimulus x with K features,

$$x_i \in C_i$$

for $i \in [1, ..., K]$, where C_i is the set (of size n_i) of all possible values for feature i. Using the equality function, we implement an indicator function,

$$[i = j] = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

for all values of all features.

In total, there are $M = \prod_i n_i$ possible stimuli and all of our codes are designed to produce a unique response pattern for each of these M possible stimuli. Importantly, our results do not depend on any particular distribution of these stimuli. This follows from statement 4, which shows that each stimulus has the same set of distances from the other stimuli in code response space. So, all stimuli have the same probability of decoding error – and, thus, heterogeneity in their probability of occurrence will not affect the overall probability of a decoding error.

2.5.2 Definition of the codes

Our definition for nonlinear mixed selectivity follows that given in [62]. We describe it with some generalizations here.

The codeword c corresponding to stimulus $x \in X$ is produced by

$$c(x) = \beta t_O(x)$$

where β is a matrix of size $N \times D$ and $t_O(x)$ is the encoding function of order O. Our codes will primarily be differentiated by $t_O(x)$, while β will be used to equalize their representation energy V and population size N (see *Linear transform (β)* in *Methods*).

The elements of the vector $t_O(x)$ are products of indicator functions, and therefore can only be either one or zero. In particular, for order O, the vector $t_O(x)$ has length corresponding to the number of valid feature-value indicator functions of size O – and each element of $t_O(x)$ corresponds to one of those combinations. More formally, for $A \in G_K^O$, where G_K^O is the set of all combinations of O elements from $[1, ..., K]$, and $(i, ..., j) \in (C_{A_1}, ..., C_{A_O})$,

$$t_O(x)_k = \left[x_{A_1} = A_1^i\right] ... \left[x_{A_O} = A_O^j\right]$$

with individual neurons (indexed by k) corresponding to all feature combinations A and all value combinations for those features $(i, ..., j)$.

Thus, $1 \leq O \leq K$, where K is the total number of stimulus features, and all codes with $O \geq 2$ are mixed while codes with $O = 1$ are pure codes, following [62]. We will use the term "neuron" to refer to coding units in our models and simulations as well as to refer to biological neurons in the brain to make their analogous roles clear. In our formulation, both mixed and pure codes will always have complete coverage; that is, there will be a neuron coding for every feature value or possible combination of feature values and each of the M stimuli will have a corresponding unique codeword.

Finally, in the main text and much of the methods, we will often make the assumption that all stimulus features take on the same number of values – that is, that $n_i = n$ for all $i \in [1, ...K]$. While this assumption does change the population size of our codes, it does not change their minimum distance or representation energy (as can be seen below). Thus, it has a negligible effect on our results.

Code example

For $K = 3$ and $n = 2$, under our formalization there are codes of three different orders that code for the n^K stimuli.

O = 1: This code has nK neurons and below we give some example stimuli (on the left, with the three features each taking on one of their two possible values, 1 or 2) and codewords (across the activity of the neurons, on the right):

111	1	0	1	0	1	0
211	0	1	1	0	1	0
122	1	0	0	1	0	1
222	0	1	0	1	0	1

Note that for each of these stimuli, there are always three neurons responding with 1. Further, the smallest distance between any two codewords is $\sqrt{2}$, between 111 and 211 as well as 122 and 222. This is of course not the smallest number of neurons that we could use to represent the set of 8 stimuli. The smallest number of neurons that could represent these stimuli is $\log_2 n^K = \log_2 8 = 3$ neurons, which could use a representation similar to the one we have used to represent the stimuli on the left hand side of the above table.

Thus, this encoding strategy has added redundancy to our representation of the stimuli.

O = 2: This code has $\binom{K}{O} n^O = \binom{3}{2} 2^2 = 12$ neurons. It can be viewed as three separate $O = 2$ codes for the three different size 2 subsets of the 3 features. We make that explicit in our example:

111	1	0	0	0	1	0	0	0	1	0	0	0
211	0	1	0	0	1	0	0	0	0	1	0	0
122	0	0	1	0	0	0	0	1	0	0	1	0
222	0	0	0	1	0	0	0	1	0	0	0	1

Note that any two of these three subpopulations alone would produce a code with unique codewords for each of the stimuli. However, they would preferentially represent one of the three features and cause errors to be more likely for the other two features. The minimum distance between any of the stimuli is now 2 and the number of neurons active is 3.

O = 3: This code has $n^K = 8$ neurons, that each code for a unique combination of the three features – and therefore for a unique stimulus. As in:

111	1	0	0	0	0	0	0	0
211	0	1	0	0	0	0	0	0
122	0	0	0	0	0	0	1	0
222	0	0	0	0	0	0	0	1

Note that there is now only one neuron active for each stimulus, and the minimum distance is $\sqrt{2}$.

Next, we formalize these properties: population size, minimum distance, and representation energy (or the number of active neurons) and derive expressions for each of them for general K and n.

2.5.3 Code properties

Population size (D_O) of the codes

The population size of a code is the length of $t_O(x)$ for that code. Since we know that a code of order O will have an element for each possible combination of feature-values of size

O, the length of the vector can be framed as a counting problem:

$$D_O = \sum_{A \in G_K^O} \prod_{i \in A} n_i$$

where G_K^O is the set of all subsets of $[1, ..., K]$ with size O and $n_i = |C_i|$. This expression is somewhat cumbersome, so, as described above, we assume that $n = n_j$ for all $j \in [1, ..., K]$. This gives,

$$D_O = \binom{K}{O} n^O$$

where $\binom{K}{O}$ is the binomial coefficient, defined as

$$\binom{n}{r} = \frac{n!}{(n-r)!r!}$$

if $n \geq r$, otherwise $\binom{n}{r} = 0$.

For $O = 1$ (the pure code), the population size is

$$D_1 = Kn$$

and, for $O = K$ (the fully mixed code), it is

$$D_K = n^K$$

Thus, the population size (i.e., the length of the vector) grows exponentially with the order of the code.

Representation energy (P_O) of the codes

We quantify the amount of energy that each coding scheme uses to transmit codewords. In particular, we will model energy in two ways and will see that these are equivalent for large n_i and do not substantially change our results for smaller n_i.

Sum of variance across dimensions: Taking the variance of a particular dimension as the energy used by that dimension for coding, we can simply take the sum across all of the dimensions. With the definition of variance,

$$\mathrm{Var}(X) = \mathrm{E}\left(X^2\right) - \mathrm{E}(X)^2$$

we can express representation energy (P_O) as:

$$\begin{aligned} P_O &= \sum_{i}^{D_O} \mathrm{Var}(t_O(x)_i)_X \\ &= \sum_{A \in G_K^O} \prod_{i \in A} n_i \left[\frac{1}{\prod_{i \in A} n_i} - \frac{1}{\prod_{i \in A} n_i^2} \right] \\ &= \sum_{A \in G_K^O} \left[1 - \frac{1}{\prod_{i \in A} n_i} \right] \\ &\leq \binom{K}{O} \end{aligned}$$

where G_K^O is the set of all subsets of size O of K elements (here, features).

With large n_i the second term in the sum becomes very small, and we can see that the upper bound of the last line gives a good approximation of the representation energy.

So, for $O = 1$,

$$P_1 \approx K \tag{2.4}$$

For, $O = K$,

$$P_K \approx 1 \tag{2.5}$$

Squared distance: We also notice that for a code of a particular order, all of the codewords have the same distance from the zero-activity state (the origin). This distance provides a different notion of energy consumption, through $P_O = w_O^2$, where w_O is the distance for a code of order O. Formally, it differs from the notion of energy consumption given above only in that the squared mean activity is not subtracted. That is,

$$P_O = w_O^2 = \sum_i^D \mathrm{E}\left(t_i(x)^2\right)_X$$

rather than

$$P_O = \sum_i^D \mathrm{E}\left(t_i(x)^2\right)_X - \sum_i^D \mathrm{E}\left(t_i(x)\right)_X^2$$

Following the derivation above, the distance is:

$$w = \left[\binom{K}{O}\right]^{\frac{1}{2}} \tag{2.6}$$

and the representation energy (or squared distance) is

$$P_O = \binom{K}{O}$$

and, for $O = 1$,

$$P_1 = K$$

and, for $O = K$,

$$P_K = 1$$

That is, this gives the same answer as our other measure for energy, but does not depend on the assumption that the n_i are large.

Use of either of these two measures does not substantively affect our results. In our simulations, we will use the former because it slightly benefits pure codes (because the mean activity of neurons in pure codes is generally higher than that in mixed codes, so there is a larger reduction in their representation energy by the subtraction of the squared mean) and we are exploring the benefits of mixed codes.

Minimum distance (Δ) of the codes

The smallest distance between any two codewords is directly related to the probability that a decoder will make an error when attempting to discriminate between those two codewords, and can be used to bound the performance of decoders in general.

Intuitively, the minimum distance will be between stimuli that differ by only one feature. In particular, the minimum distance will be dependent on how many of the $\binom{K}{O}$ combinations

of O features contain the feature that differs between the two stimuli. This is given by $\binom{K-1}{O-1}$ (which counts the ways one can select the rest of the $O-1$ features, assuming the differing feature is already included in the combination), and figures prominently in our equation for minimum distance below. We also develop an equation for the distance between stimuli that differ by more than one feature and show that this distance is increasing in the number of features two stimuli differ by, see *Code distances* in *Supplement*. These two steps show that this intuition about the minimum distance is correct.

By statement 2, we know that the minimum value of $d(K, O, v)$ occurs when $v = 1$. We can then evaluate our expression for distance, found in statement 1, at $v = 1$

$$\begin{aligned} d(K, O, 1) &= \left[2 \sum_{i}^{1} \binom{1}{i} \binom{K-1}{O-i} \right]^{\frac{1}{2}} \\ &= \left[2 \binom{1}{1} \binom{K-1}{O-1} \right]^{\frac{1}{2}} \\ \Delta_O &= \left[2 \binom{K-1}{O-1} \right]^{\frac{1}{2}} \end{aligned}$$

Now, we can evaluate this expression for any K and O that we desire. For $O = 1$,

$$\Delta_1 = \sqrt{2}$$

and, for $O = K$,

$$\Delta_K = \sqrt{2}$$

While the minimum distance for these two codes is the same, their representation energy is different (see Eq. 2.4 and Eq. 2.5). Further, minimum distance and power are both weakly unimodal around $O = \lfloor K/2 \rfloor$ (Figure 2.1E, center and right).

2.5.4 Minimum distance-representation energy ratio

A straightforward way to describe code performance in a single number is to take the ratio between squared minimum distance and representation energy. Codes with larger ratios will typically have a lower probability of decoding error given the same noise level.

$$\begin{aligned}\frac{\Delta^2}{P} &= 2\frac{\binom{K-1}{O-1}}{\binom{K}{O}} \\ &= 2\frac{(K-O)!O!(K-1)!}{(K-O)!(O-1)!K!} \\ &= 2\frac{O}{K}\end{aligned} \tag{2.7}$$

which is strictly increasing with order (Figure 2.1F, left).

2.5.5 Linear transform (β)

In comparing codes of different orders, it useful to give the codes the same representation energy (P_O) and population size (D_O), so that it is clear that the differences in performance are due to the different effective dimensionalities and arrangements of stimulus representations produced by codes of different orders, rather than differences in representation energy or population size. Thus, we apply a linear transform β to the codewords $t_O(x)$, which sets representation energy to be equal to some value V and population size to be equal to some value N, which are both now decoupled from code order, O. Thus, we can flexibly compare codes of different orders given the same energy – or, put another way, using β, codes of different orders can be implemented with arbitrary representation energy and relatively unconstrained population sizes (though, $D_O \leq N$). In practice, β is an $N \times D_O$ matrix.

This step also has an important conceptual interpretation. In neural recordings, the activity of large populations of neurons have been found to inhabit a subspace of all possible neural

responses with much lower dimensionality than the number of neurons (i.e., the maximum possible dimensionality, if all neurons were independent). In our framework, this subspace can be viewed as the D_O-dimensional space of the codewords, while a neural implementation of the code may use N neurons (with $N \geq D_O$). Through the transform from the D_O-dimensional codeword space to the N-dimensional response space, β can be used to make several quantitative and qualitative changes to the final representation. We will summarize the ones that are most relevant here:

1. It can expand ($P_O < V$) or contract ($P_O > V$) the representation by either increasing or decreasing the representation energy of the code.

2. It can perform a rotation or reflection of the codewords, which can both change the sparsity of the neural response as well as move from the strict binary representation of the codewords to a more graded representation, where neurons can have different non-zero firing rates for different stimuli. For instance, each element of the linear transform matrix can be sampled from a Normal distribution with zero mean (and normalized according to the conditions set out below). Then, each neuron in the population would have a non-zero response to every stimulus with high probability.

3. It can project the codewords into a higher-dimensional space, as when $D_O < N$ – in this way, there can be a tradeoff made between having fewer neurons with high individual SNRs or more neurons with lower individual SNRs, to result in the same population-level SNR (which is the SNR used in the rest of the manuscript). For example, in this way, a single neuron with a high individual SNR could be replaced by two neurons with lower individual SNRs but the same feature tuning without affecting the decoding performance of the code. Thus, the population size N of the code would be larger, but both the effective dimensionality and code performance would not be affected.

Thus, through the linear transform, the population representation of the stimuli can be made more realistic and heterogeneous. In addition, as shown in Figure 2.1G, the sparsity of our codewords increases with code order. Through choice of β, that dependence can be, in part, broken.

Importantly, only the change in representation energy due to the linear transform will alter code performance (see below). Here, we only consider linear transforms that scale all of the codeword dimensions uniformly – and, as a consequence, maintain the uniform representation energy across individual codewords. This class of linear transforms cannot change the relative geometry of the codeword representations, it can only rescale, rotate, and embed it. This relative geometry is what produces the increase in code performance with code order at the same representation energy that we describe here.

Heterogeneous rescaling of the codeword dimensions by either a static or dynamic linear transform is one way in which a particular code could be optimized for the representation of a non-uniform stimulus distribution. We consider this possibility further in the Discussion.

Selecting a linear transform (β)

In choosing β, we must satisfy four constraints:

1. $N \geq D_O$
2. $\beta^\dagger \beta = I$, where I is the $D_O \times D_O$ identity matrix and $\beta^\dagger$ is the pseudoinverse of β.
3. The vector length, H, of each column in β must be the same.
4. $\mathrm{E}(\beta_{ij}\beta_{ik})_{j \neq k} = 0$; this will be true, for instance, for β where the rows or columns are sampled from an independent Normal distribution.

Even given these constraints, there is significant flexibility in the choice of β, which allows β

to be used to alter some of the qualitative features of our codes, as described above. However, as mentioned above and as we will show in more detail below, it is only the vector length of the columns of β, H, that affects the performance of the code. As a result, throughout our simulations, we will use βs that are proportional to the identity matrix for simplicity and ease of interpretation.

For β with length H, the transformed code $\beta t_O(x)$, where $t_O(x)$ has representation energy P_O, will have representation energy

$$V = H^2 P_O \tag{2.8}$$

We derive this using the squared distance definition of energy. So, the energy of the original code $t_O(x)$ is given by,

$$P_O = \binom{K}{O}$$

After applying β, we want to find the square of the average distance of the codewords from the origin, or V, under the squared distance definition of energy.

So, we want to find, where $c(x) = \beta t_O(x)$, X is the set of stimuli, and M is the number of

stimuli,

$$
\begin{aligned}
V &= E\left[\sum_{i}^{N}\left(c_i(x)\right)^2\right] \\
&= \frac{1}{M}\sum_{x\in X}\sum_{i}^{N}\left(c_i(x)\right)^2 \\
&= \frac{1}{M}\sum_{x\in X}\sum_{i}^{N}\left(\sum_{j\in D_x}\beta_{ij}\right)^2 \\
&\quad \text{where } D_x \text{ is the set of non-zero indices of } t_O(x) \text{ for } x \\
&= \frac{1}{M}\sum_{x\in X}\sum_{i}^{N}\sum_{j\in D_x}\beta_{ij}^2 \\
&\quad \text{by the definition of } \beta\text{, constraint 4} \\
&= \frac{1}{M}\sum_{x\in X}\sum_{j\in D_x}H^2 \\
&\quad \text{by the definition of } \beta\text{, constraint 3} \\
&= \frac{1}{M}\sum_{x\in X}\binom{K}{O}H^2 \\
&= H^2\binom{K}{O} \\
&= H^2 P_O
\end{aligned}
$$

Thus, we can give different codes the same representation energy V by choosing $H = \sqrt{V/P_O}$ for each O.

The effect of β on minimum distance

The distance between two points $c_i = \beta t_O(x_i)$ and $c_j = \beta t_O(x_j)$, represented as d_{ij}^{β}, is given by

$$d_{ij}^{\beta} = H d_{ij} \tag{2.9}$$

where d_{ij} is the distance between the two points $t_O(x_i)$ and $t_O(x_j)$.

We know that points x_i and x_j are both $\sqrt{P}$ units away from the origin while codewords c_i and c_j are $H\sqrt{P}$ units from the origin (by Eq. 2.8 and equation 2.6). We want to find d_{ij}^{β}.

The angle between the two points is

$$\theta = \sin^{-1} \frac{\frac{1}{2} d_{ij}}{\sqrt{P}}$$

so, we can rearrange to find:

$$\begin{aligned} d_{ij}^{\beta} &= 2H\sqrt{P} \sin\theta \\ &= 2H\sqrt{P} \frac{\frac{1}{2} d_{ij}}{\sqrt{P}} \\ &= H d_{ij} \end{aligned}$$

It follows directly from Eq. 2.9 that the minimum distance after β is applied, δ, is given by

$$\delta = H\Delta$$

Further, it follows that the ratio given in Eq. 2.7 is not altered by H, or choice of particular

β, since

$$\frac{\delta_O^2}{V_O} = \frac{H^2\Delta_O^2}{H^2 P_O} = \frac{\Delta_O^2}{P_O}$$
$$= 2\frac{O}{K}$$

2.5.6 Full channel details

We simulated codes of all possible orders for particular choices of K and n. Three important choices were made for these simulations. First, the codewords from each code were passed through a linear transform β. The linear transform was used to equate the population size and representation energy of different order codes, such that we could investigate code performance when each order of code had the same number of participating units and the same signal-to-noise ratio (SNR $= \sqrt{V/\sigma^2}$ where V is the code representation energy after the linear transform is applied and σ^2 is the noise variance), as in Figure 2.2 and see *Linear transform (β)* in *Methods*. Second, the noise in the channel was chosen to be additive and to follow an independent Normal distribution across code dimensions. Third, we use maximum likelihood decoding (MLD) to estimate the original stimulus. This choice is consistent with Bayesian and probabilistic formulations of neural encoding and decoding[142–144]. While inclusion of noise correlations would be an interesting topic for future research, we show here that they are not essential for any performance increases due to nonlinear, conjunctive mixing.

Code availability

All of the code for the simulations was written in Python (3.6.4) using NumPy (1.14.2), SciPy (1.0.1)[145], and Scikit-learn (0.18.1)[146]. The code is available on github. For each SNR and each code order, 5000 to 10000 trials were simulated.

2.5.7 Estimating the error rate

While the minimum distance-representation energy ratio we derive in Eq. 2.7 provides useful insight into the performance of codes of different orders, it does not give a direct estimate of the probability of decoding error. In particular, it is difficult to interpret the magnitude of performance differences without incorporating the magnitude of the noise itself, the decoder used, and the arrangement of all of the codewords in coding space to estimate error rate directly. Here, we incorporate the details of the full channel to directly estimate the error rate via a union bound estimate (UBE).

That is, with the channel,

$$\begin{aligned} r(x) &= c(x) + \eta \\ &= \beta t_O(x) + \eta \end{aligned}$$

where $\eta \sim \mathrm{N}(0, \sigma^2)$ (see Figure 2.1A for a schematic) and a maximum likelihood decoding function f such that $\hat{x} = f(r(x))$ where $\hat{x}$ is the maximum likelihood estimate of x given $r(x)$, we want to estimate the probability that $\hat{x} \neq x$ across X – that is, the probability of

decoding error, PE. To begin,

$$\begin{aligned}
\text{PE} &= \sum_{x \in X} p(x) P\left(\cup_{x \neq a \in X} \hat{X} = a | X = x\right) \\
&= P\left(\cup_{x \neq a \in X} \hat{X} = a | X = x\right) \\
&\text{by statement 4} \\
&= \sum_{x \neq a \in X} P(\hat{X} = a | X = x) \\
&\text{by the disjoint nature of decoding events} \\
&\leq \sum_{x \neq a \in X} Q\left(\frac{d_{\text{E}}(x, a)}{2\sigma}\right) \qquad (2.10)
\end{aligned}$$

where $Q(y)$ is the ccdf at y of $\mathcal{N}(0, 1)$ and $d_{\text{E}}(x, y)$ is the Euclidean distance between the codewords corresponding to x and y (i.e., the Euclidean distance between $\beta t_O(x)$ and $\beta t_O(y)$).

We can proceed further by using the function:

$$d(K, O, v) = \left[2 \sum_{i}^{v} \binom{v}{i} \binom{K - v}{O - i}\right]^{\frac{1}{2}}$$

which gives the distance between two stimuli that differ by v out of K total features in an order O code (see *Code distances* in *Supplement* for a derivation), and the fact that the number of stimuli that differ by v features from a particular stimulus is given by

$$N_{\text{all}}(v) = \binom{K}{v} (n - 1)^v$$

Thus, Eq. 2.10 can be rewritten as a sum of all stimuli $x \neq a$ arranged by their distance

from x (the original stimulus):

$$PE \leq \sum_{x \neq a \in X} Q\left(\frac{d_{\mathrm{E}}(x,a)}{2\sigma}\right) = \sum_{v=1}^{K} N_{\mathrm{all}}(v)\, Q\left(\frac{H d(K,O,v)}{2\sigma}\right)$$

where $H = \sqrt{V/P_O}$, due to the linear transform

$$= \sum_{v=1}^{K} N_{\mathrm{all}}(v)\, Q\left(\sqrt{\frac{V}{P_O}}\frac{d(K,O,v)}{2\sigma}\right)$$

$$= \sum_{v=1}^{K} N_{\mathrm{all}}(v)\, Q\left(\frac{\mathrm{SNR}}{2\sqrt{P_O}}\, d(K,O,v)\right)$$

This expression provides an explicit upper bound that well-approximates our simulation results (Figure 2.2A) and we use this expression to characterize code performance in Figure 2.2B. However, it is difficult to gain intuition about code performance through this expression. Thus, we reformulate the sum to include only the terms that give the likelihood of errors to stimuli at minimum distance. This works as an approximation because these errors require the smallest noise and are therefore, in most cases, exponentially more likely than errors to stimuli at even the next smallest distance. Using our expression for minimum distance and for the number of stimuli at that distance for each code:

$$\mathrm{PE} \leq \sum_{v=1}^{K} N_{\mathrm{all}}(v)\, Q\left(\frac{\mathrm{SNR}}{2\sqrt{P_O}}\, d(K,O,v)\right)$$

$$\approx N_{\Delta}(O) Q\left(\frac{\mathrm{SNR}}{2\sqrt{P_O}}\, d(K,O,1)\right)$$

$$= N_{\Delta}(O) Q\left(\frac{\mathrm{SNR}}{2}\,\frac{\Delta_O}{\sqrt{P_O}}\right)$$

$$= N_{\Delta}(O) Q\left(\frac{\mathrm{SNR}}{\sqrt{2K/O}}\right)$$

where $N_{\Delta}(O)$ is the number of neighbors at minimum distance for the code of order O, derived in *Code neighbors* in *Supplement*. Thus, we can see that PE depends most strongly

on the minimum distance-representation energy ratio and SNR. Further, for $O < K$, this approximation is strictly decreasing with order, implying the main result of our paper: that increasing mixing (O) increases code reliability. This is matched by our simulation results (Figure 2.2A). Further, in the full approximation above, the $O = K$ code is guaranteed to have the smallest error rate, due to the fact that all of its stimulus representations are at maximum distance from each other ($d(x, a) = \sqrt{2V}$, derived in *Code distances* in *Supplement*), while all other codes have at least some proportion of stimuli that are closer together (and therefore are more likely to give rise to errors). This is also matched by our simulation results (Figure 2.2A), though, for large K, the performance of high-order codes becomes increasingly similar (Figure 2.2A, bottom and Figure 2.2B).

2.5.8 Total energy

Similar to [129], we assume that all neurons, whether spiking or not, consume some baseline, non-zero amount of energy – due to passive maintenance processes, including the circulation of ion channels, and due to spontaneous activity. We define this amount of energy to be equal to one unit. Next, we assume that spiking neurons consume the baseline energy plus an amount of energy proportional to the square of their firing activity; this activity summed across the population is the representation energy (P_O). So, the total energy consumption of a code, E, can be written:

$$E = \epsilon V + D_O \tag{2.11}$$

where ϵ controls the proportional cost of spiking relative to passive maintenance costs. This ϵ will vary between neuron types, but has been estimated by experiment to be around 10 to 10^2[129].

From Eq. 2.11, we see that a code of order O allocated E total energy would have,

$$V = \frac{E - D_O}{\epsilon}$$

and

$$\delta^2 = \frac{2O}{K\epsilon}(E - D_O)$$

where only codes with $V > 0$ (that is, $E > D_O$) can be implemented in practice. This δ is used in the comparisons for Figure 2.2d. From this expression, we observe that the particular value of ϵ does not change the relative performance of codes with different orders. So, our results in Figure 2.2d do not depend on ϵ.

Further, we find that when $\delta_O = \delta_{O+1}$ as a function of E to discover when the $O + 1$-order

code will begin to outperform the O-order code:

$$\begin{aligned}
\delta_O^2 &= \delta_{O+1}^2 \\
\frac{2O}{K\epsilon}\left(E - D_O\right) &= \frac{2(O+1)}{K\epsilon}\left(E - D_{O+1}\right) \\
O\left(E - D_O\right) &= (O+1)\left(E - D_{O+1}\right) \\
OE - OD_O &= (O+1)E - (O+1)D_{O+1} \\
E &= (O+1)D_{O+1} - OD_O \\
&= (O+1)\binom{K}{O+1}n^{O+1} - O\binom{K}{O}n^O \\
&= (K-O)\binom{K}{O}n^{O+1} - O\binom{K}{O}n^O \\
&= (K-O)\binom{K}{O}n^{O+1} - \frac{O}{n}\binom{K}{O}n^{O+1} \\
&= \frac{nK-(n+1)O}{n}\binom{K}{O}n^{O+1} \\
&= (nK-(n+1)O)\binom{K}{O}n^O \\
E_{O\to O+1} &= (nK-(n+1)O)\,D_O
\end{aligned}$$

and using this for $O = 1$, we find that

$$\begin{aligned}
E_{\text{mixed}} &= n^2K^2 - n^2K - nK \\
&< n^2K^2
\end{aligned} \tag{2.12}$$

such that for $E > E_{\text{mixed}}$ a mixed code (i.e., a code of order $O > 1$) will always provide better performance than a pure code.

2.5.9 Experimental details and task description

We used experimental data in Figure 2.4 that was previously published in a separate study [135]. The full methods are given in the original paper, though we briefly review several key points here. The data may be requested from the authors of the previous study.

The behavioral task

See the schematic in Figure 2.4B. First, a moving dot stimulus in a direction that was on one side of a learned category boundary was presented while the animal fixated. Then, there was a delay period during which the animal was compelled to saccade to one of two locations before, finally, a second motion stimulus was presented and the animal reported whether the category of the first (or sample) stimulus matched the category of the second (or test) stimulus. The division of the 360° of motion direction into two contiguous categories was arbitrary, and learned by the animals over extensive training.

The electrophysiological recordings and analysis

The experimenters recorded from 64 lateral intraparietal area (LIP) neurons in two monkeys (monkey J: $n = 35$; monkey M: $n = 29$) during performance of the DMC task. Recordings were performed using single 75 µm tungsten microelectrodes (FHC). Units were sorted offline, and selected for quality and stability. No information about the LIP subdivision from which each neuron was collected is available.

Linear models for motion category (category 1 or 2) and saccade direction (toward or away from the neuronal RF) with interaction terms (between category and saccade direction) were fit using an L1 prior in scikit-learn[146] (i.e., the Lasso fitting procedure) to all neurons with greater than 15 trials for each of the four conditions (61/71 neurons). The data used for fitting was subsampled without replacement so that each condition had the same number

of trials as the condition with the fewest recorded trials (e.g., if there were 40, 35, 24, and 37 from each condition for a single neuron, then 24 trials would be subsampled from each group for the fitting). Fit coefficients were tested for significance via a permutation test (using 5,000 permutations) at the $p < .05$ level after applying a Bonferroni correction for multiple comparisons. Spikes were counted in the 20 ms to 170 ms window after the saccade was made and then spike counts for each neuron were z-scored across the four conditions.

2.6 Supplemental Information

"Nonlinear mixed selectivity supports reliable neural computation"

W. Jeffrey Johnston, Stephanie E. Palmer, David J. Freedman

2.6.1 Glossary of terms

M	The number of stimuli transmitted by a code.
Δ_O	The minimum distance of the code of order O.
δ_O	The minimum distance of a code of order O after β is applied.
P_O	The representation energy used by a code of order O.
V	The representation energy used by a code after β is applied.
D_O	The population size of a code of order O.
N	The population size of the code after β is applied.
K	The number of features that a stimulus has.
C_i	The set of values that feature i can take on.
n_i	The size of set C_i; that is, $n_i = \|C_i\|$
G_K^s	The set of all possible subsets of $[1, ..., K]$ with size s; $\{X \subset [1, ..., K] : \|X\| = s\}$
x	A stimulus; a vector of length K, where $x_i \in C_i$ for all i.
$t_O(x)$	The encoding function of order O. It takes a stimulus (x) and produces the representation of that stimulus in a code of order O – also referred to as the codeword. The representation is a vector of length D_O of ones and zeros.
β	The amplifying transform. It is applied to the codeword ($t_O(x)$) and produces the amplified encoding; β is a matrix of size $N \times D$ and must satisfy the constraints given in *Linear transform (β)* in *Methods*.
H	The power in each column of β; $\sqrt{\sum_i^N \beta_{ij}^2} = H$ for all j.

η	A noise term. Here, always Gaussian, with $\eta \sim \mathrm{N}(0, \sigma^2)$.
$c(x)$	The amplified codeword corresponding to a given stimulus, $c(x) = \beta t_O(x)$. It is a vector of length N.
$r(x)$	The noisy amplified codeword corresponding to a given stimulus, $r(x) = c(x) + \eta$. It is a vector of length N.
$f(r)$	The maximum likelihood decoding function for a particular code. It solves the equation $\mathrm{argmax}_x\, P(r\|x)P(x)/P(r)$.
$\hat{x}$	The estimate of x, derived from a noisy representation, $\hat{x} = f(r)$.

2.6.2 Code distances

We develop some general properties of the distances between stimulus representations in our codes here. These are useful in conclusively proving the minimum distance, as well as showing that each stimulus has the same neighbor structure as all the other stimuli in a particular code.

Statement 1. *The distance between two stimulus codewords is given by*

$$d(K, O, v) = \left[2 \sum_{i}^{v} \binom{v}{i} \binom{K-v}{O-i} \right]^{\frac{1}{2}}$$

where v is the number of features the stimuli differ in, O is the order of the code, and K is the number of features.

Derivation. Using the set G_K^O with $|G_K^O| = \binom{K}{O}$, we see that when we change a feature $i \in [1, ..., K]$, by the definition of the indicator function and of our codes, we know that one term (a product of indicator functions) in each feature combination that includes i will flip from 0 to 1 and another term will flip from 1 to 0. Thus, given the subset $B_i^O = \{b \in G_K^O | i \in b\}$,

we obtain a distance of $\sqrt{2|B_i^O|}$ from changing the value of feature i. When we change a second term, j, we obtain $B_j^O = \{b \in G_K^O | j \in b\}$. The distance between the two stimuli is then related to the size of the union of these two sets: $\sqrt{2|B_i^O \cup B_j^O|}$.

So, to find the distance between two codewords, we need to count the number of features in which they differ and then find the distance, given the order of the code O and the number of stimulus features K.

$$\begin{aligned} d(K, O, v) &= \left[2 \left| \bigcup_i^v B_i^O \right| \right]^{\frac{1}{2}} \\ &= \left[2 \sum_i^v \binom{v}{i} \binom{K-v}{O-i} \right]^{\frac{1}{2}} \end{aligned}$$

where the second binomial coefficient counts the number of subsets containing exactly i of the v changed features and the first binomial coefficient counts the number of different ways i features could be chosen from the v changed features. Since our codes include all combinations, the identities of the features changed does not matter – only the number of them. □

Next, it will be useful to know that this distance function is increasing with v, as, combined with statement 1, it will allow us to find the minimum distance.

Statement 2. *The function $d(K, O, v)$ is increasing with v.*

Derivation. We want to show that $d(K, O, v) \leq d(K, O, v+1)$.

$$\begin{aligned} 0 &\leq d(K, O, v+1)^2 - d(K, O, v)^2 \\ &= \left| \bigcup_i^{v+1} B_i^O \right| - \left| \bigcup_i^{v} B_i^O \right| \\ &= \left| B_{v+1}^O \setminus \bigcup_i^{v} B_i^O \right| \end{aligned}$$

where the last line is the size of the set of values that are in B_{v+1}^O and not in any of the other B_i^O for $i \in [1, ..., v]$. The relationship holds because a set cannot have a negative size. Thus, $d(K, O, v+1) \geq d(K, O, v)$ and therefore the function d is increasing in v. □

Finally, we will derive the maximum distance between any two codewords in a code. Intuitively, this will be when none of the same neurons are active for the two codewords. We can see this from our equations above by noticing that $d(K, O, v)$ has a (potentially non-unique) maximum at $v = K$ (by statement 2) and

$$\begin{aligned} d(K, O, K) &= \left[2 \left| \bigcup_i^{K} B_i^O \right| \right]^{\frac{1}{2}} \\ &= \left[2 \left| G_K^O \right| \right]^{\frac{1}{2}} \\ &= \left[2 \binom{K}{O} \right]^{\frac{1}{2}} \\ &= [2P_O]^{\frac{1}{2}} \end{aligned}$$

After the linear transform, this becomes $\sqrt{2V}$, and therefore does not depend on code order. Finally, we identify this maximum distance as equivalent to the minimum distance of the

$O = K$ code after application of the linear transform:

$$\begin{aligned}\delta_K &= \sqrt{\frac{V}{P_K}}\Delta_K \\ &= \sqrt{V 2\frac{K}{K}} \\ &\text{by Eq. 2.7} \\ &= \sqrt{2V}\end{aligned}$$

which demonstrates that all stimulus representations in the $O = K$ code are at maximum distance from each other, by statement 2.

2.6.3 Code neighbors

For the UBE, it becomes necessary to know the number of codewords at minimum distance from any given codeword ($N_\Delta(O)$).

Statement 3. *The number of neighbors at minimum distance for a code of order O* $N_\Delta(O)$ *is given by:*

$$N_\Delta(O) = \begin{cases} K(n-1) & O < K \\ n^K - 1 & O = K \end{cases} \tag{2.13}$$

Derivation. From the fact that the distance function is increasing with v (statement 2), we know that $d(K, O, 1)$ is the minimum of $d(K, O, v)$, but it may or may not be a unique minimum.

Thus, we want to find O such that $d(K,O,1) < d(K,O,2)$,

$$
\begin{aligned}
0 &< d(K,O,2)^2 - d(K,O,1)^2 \\
&= \binom{2}{2}\binom{K-2}{O-2} + \binom{2}{1}\binom{K-2}{O-1} - \binom{1}{1}\binom{K-1}{O-1} \\
&= \binom{K-2}{O-2} + 2\binom{K-2}{O-1} - \binom{K-1}{O-1} \\
&\text{exploiting binomial identities to make all binomial terms equal} \\
&= \binom{K-2}{O-2} + 2\frac{K-2+1-O+1}{O-1}\binom{K-2}{O-2} - \frac{K-1}{O-1}\binom{K-2}{O-2} \\
&= \binom{K-2}{O-2} + 2\frac{K-O}{O-1}\binom{K-2}{O-2} - \frac{K-1}{O-1}\binom{K-2}{O-2} \\
&= \left[1 + 2\frac{K-O}{O-1} - \frac{K-1}{O-1}\right]\binom{K-2}{O-2} \\
&= \frac{O-1+2K-2O-K+1}{O-1}\binom{K-2}{O-2} \\
&= \frac{K-O}{O-1}\binom{K-2}{O-2} \\
&\text{this is undefined for } O=1\text{, which is undesirable} \\
&= \frac{K-1}{K-1}\frac{K-O}{O-1}\binom{K-2}{O-2} \\
0 &< \frac{K-O}{K-1}\binom{K-1}{O-1}
\end{aligned}
$$

This last expression is true when $1 \leq O < K$ and false otherwise (i.e., when $O = K$). When it is true, it implies that changing one stimulus feature produces codewords at a closer distance than changing two stimulus features. Now, we must find how many stimuli differ by a single feature from a given stimulus. Any single feature of the K features could be changed, and it could be changed to any one of $n-1$ different values (excluding its current value) – so, $N_\Delta(O) = K(n-1)$ for $O < K$.

If $O = K$, then $G_K^K = \{\{1, ..., K\}\} = B_1^K$ and since B_i^K cannot grow beyond the size of G_K^O, all codewords must be at the same distance. Thus, $N_\Delta(O) = n^K - 1$ for $O = K$. □

Statement 4. *The number of neighbors at a fixed distance does not depend on codeword identity.*

Derivation. We assume that the number of neighbors at a fixed distance does depend on codeword identity and show that this leads to a contradiction. We know that codeword distance does not depend on original codeword identity (statement 1), but does depend on the number of features that the stimuli differ by. Thus, for a set of codewords to have more neighbors at a particular distance than a different set of codewords, the corresponding set of stimuli must be able to differ in more ways from the corresponding set of other stimuli. Stimuli can differ by changing 1 to K of their K features to one of the $n-1$ different values for each feature C_i. For a set of stimuli to be able to differ in more ways than a different set of stimuli, that set of stimuli must have either more features or more possible values for each feature. Either of these would contradict our definition of the stimuli (see *Definition of the stimuli* in *Methods*). □

2.6.4 Sum of spikes representation energy

To this point, we have used the squared distance or variance to characterize the relationship of spiking activity across the population to metabolic energy consumption in the form of representation energy. This is following decades of literature on neural coding[147] and communication theory[148]. However, there is some evidence to suggest that a sum of spikes, or L1, representation energy metric may be more appropriate for use in the brain[149]. To gain intuition into how this different metric for metabolic energy affects our results, we perform simulations and modify our analytical approximation to use this metric. The relevant approximation is now:

$$\mathrm{PE} \leq \sum_{v=1}^{K} N_{\mathrm{all}}(v)\, Q\left(\frac{V}{P_O}\frac{d(K,O,v)}{2\sigma}\right)$$

because $H = V/P_O$ for the linear transform.

These results illustrate that, for large numbers of features K, some intermediately mixed codes, particularly with order close to 1, will provide worse performance than the pure code, but still that highly mixed codes always provide the best performance (see Figure 2.5). A further consideration of code performance with the L1 norm may be an interesting area for future research.

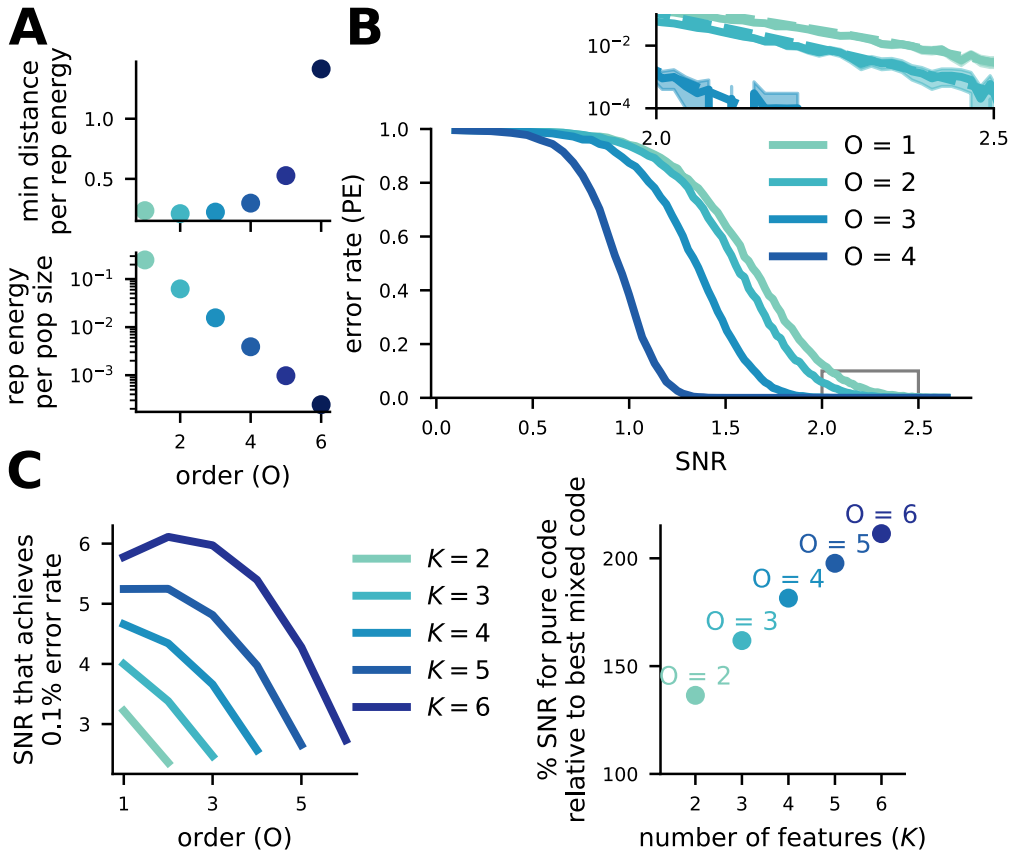

Figure 2.5: Using sum-of-spikes instead of squared distance representation energy improves the performance of higher-order codes, related to Figure 2.2. **A** (top) The minimum distance per representation energy ratio (Δ_O/P_O) for distance representation energy; and (bottom) the representation energy per population size ratio (P_O/D_O). **B** Simulation of codes with $O = 1, 2, 3, 4$ for $K = 4$ and $n = 4$. (inset) Performance of the codes relative to the approximation (dashed lines). **C** (left) Using our approximation, we show that for different K (with $n = 5$) the SNR required to reach 0.1 % decoding error has its minimum at $O = K$. (right) The representation energy required by the pure code relative to that required by the best mixed code (given by point color and label) to reach 0.1 % decoding error.

2.6.5 Alternate noise models

In the main text, we focus on additive, Gaussian noise. However, multiple other noise models have been proposed to be relevant to the brain, including Poisson, bit-flip, and input noise. We consider all of those briefly here.

Poisson and bit-flip noise

To this point, the noise in our neural channel has been Gaussian distributed, which allows us to vary the SNR down our channel independently of representation energy or firing rate. However, neural firing rates are often viewed, at least roughly, as following a Poisson process, which implies a particular SNR at different firing rates due to a strict relationship between mean firing rate and firing rate variance (though experimentally observed firing rate-SNR relationships have not followed the one expected from a Poisson process[150]). Thus, it is possible that due to the different firing rates of individual neurons used in our codes (as only the sum firing rate is held constant across codes), Poisson noise could change which code performs best.

To address this concern, we perform simulations with Poisson, instead of additive Gaussian, noise, following:

$$r(x) = f\left(\beta t_O(x)\right)$$

where $f(x)$ produces a sample from a Poisson distribution with mean x and the linear transform β is proportional to the D_O identity matrix. The results of these simulations are given in Figure 2.6A. We can see that, in this case, the qualitative performance of our codes relative to each other is not affected – and mixed codes still outperform pure codes. This is expected from previous work.

However, pure Poisson noise, modeled in this way, may not be appropriate for the nervous system. In particular, for our function $f(x)$, where $x = 0$ the result is 0 with probability 1, as is the case for a Poisson distribution. In contrast, neurons observed in the brain almost always have a non-zero spike probability due to spontaneous activity. To model this spontaneous activity, we include a baseline firing rate in our noise model, taking

$$r(x) = g\left(\beta t_O(x)\right)$$

where $g(x) = f\left(\min\left(x, r_{\text{spont}}\right)\right)$, r_{spont} is the spontaneous firing rate in the neural population, and $f(.)$ is defined as above. Thus, all neurons will have a non-zero probability of emitting noise spikes at all representation energies. The result of the simulations for these conditions are given in Figure 2.6B. Here, mixed codes still tend to perform better than pure codes. However, the $O = K$ mixed code performs worse relative to other mixed codes than with either Gaussian or pure Poisson noise.

For low representation energy (as in the shaded gray area of Figure 2.6B, where there will be only, on average .2 to 3.2 spikes of signal across the population), these Poisson-with-baseline simulations approximate the conditions of binary bit-flip noise (though the flip probability is not symmetric), and indicate that mixed codes outperform pure codes in those conditions as well.

In summary, the pattern of our results holds for numerous different response noise (i.e., channel-noise) distributions. This underlines the generality of the results derived from our three code metrics.

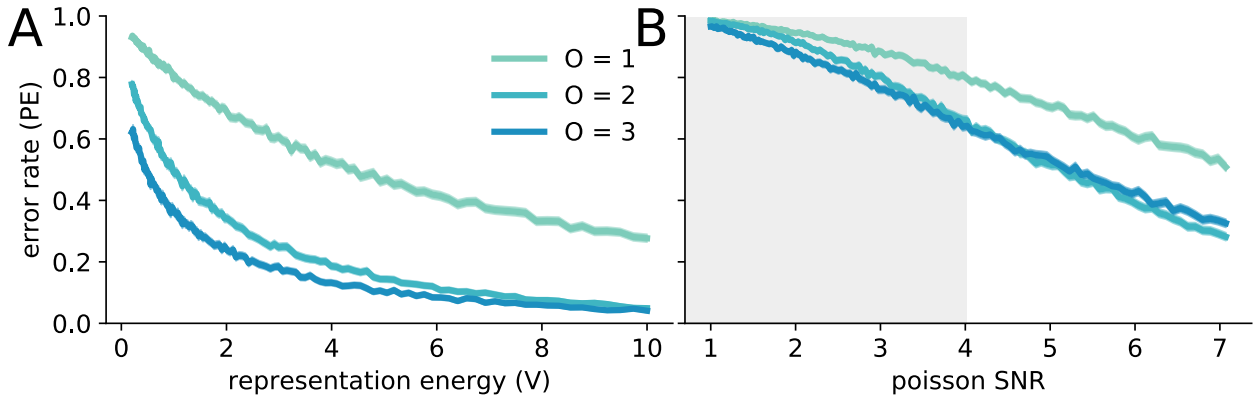

Figure 2.6: Channels with pure Poisson and Poisson-with-baseline noise have similar performance to those with Gaussian noise, related to Figure 2.2. **A** The error rate (PE) as a function of representation energy (V) for codes with pure Poisson distributed noise, $K = 3$ and $n = 5$. **B** The error rate (PE, axis same as on the left) as a function of poisson SNR for codes with Poisson-with-baseline distributed noise. Poisson SNR is defined as $\sqrt{V/r_{\text{spont}}}$, with $K = 3$, $n = 5$, and $r_{\text{spont}} = .2$. Representation energy ranges from .2 to 10, as on the left. Low values were chosen for both representation energy and r_{spont} to allow an analogue to the binary bit flip case. The gray shaded area is the region where .2 to 3.2 spikes of signal are expected across the population and few neurons will fire more than once.

Input noise

Noise in a neural system affects both the output of, as modeled in the main text and above, and the input to that system. Here, we investigate how input noise affects the robustness of codes with different levels of mixing. Previous work on mixed codes has argued that codes with more mixing are especially sensitive to input noise, and thus may make it difficult to recognize highly mixed representations of similar stimuli as similar to each other[74]. In our framework, it is true that mixed codes map similar stimuli to distant locations in response space (in part, this is what is meant by having large minimum distance, and the discrimination-generalization tradeoff discussed in [74]).

However, when the provided input is either the "true" stimulus or one of the adjacent stimuli in stimulus space (i.e., input noise is local), as would be the case if the input was from a decoder that makes local errors (e.g., with low mean squared-error), code order does not

affect robustness to input noise for decoding. This is because, while the input noise can create very different representations in response space for high-order codes, the decoder maps those different representations back to nearby areas of stimulus space, creating errors only as large as the noise in the input. This result is counter to the intuition provided by previous investigations of mixed codes with random stimuli[74]. We illustrate this without any output noise in Figure 2.7A, B.

We also simulate non-local input noise, where the input is assumed to be an $O = 1$ code stimulus representation that is subject to bit-flip noise. In this case, the $O = K$ code has the highest MSE, as expected from the previous literature, while both $O < K$ codes that we simulated have the same MSE. To explore why this is, we consider the consequences of a single input bit-flip. There are two possibilities:

1. With probability $\frac{K}{Kn} = \frac{1}{n}$, the bit-flip will change a 1 to a 0 for one of the features.

    - For an $O = K$ code, this means that none of the neurons will fire and the response without output noise will be a vector of all zeros. Thus, the decoded stimulus will be completely random with respect to the original stimulus.

    - For an $O < K$ code, only subpopulations that do not represent the bit-flipped feature will be active. The code will operate as a code of the same order on a stimulus space with $K-1$ features, and will have only $\frac{K-O}{K}$ of the representation energy of the original code. Thus, all codes will infer random values for the bit-flipped feature, and will encode the rest of the values according to a code of this nature, which will lead to reduced performance for higher order codes (though this reduction is partly corrected by the greater reliability of those codes as in Figure 2.7D).

2. With probability $1 - \frac{1}{n}$, the bit-flip will change a 0 to a 1 for one of the features. For

all codes, this will result in a second codeword becoming equally likely in our decoder, and lead to a 50 % chance of error due to this input perturbation. It will also increase the representation energy used by the code.

In simulations of codes with $K = 3$ and $n = 5$, we see that the input bit-flip noise produces a base mean squared-error even without any output noise (Figure 2.7C), due to the effects described above. The $O = 1$ and $O = 2$ codes have equivalent performance, while the $O = 3 = K$ code performs worse (again, following the pattern described above). However, when we simulate the full channel over a variety of SNRs at a fixed input bit-flip probability (Figure 2.7D), code performance replicates the broad trends of our MSE analysis in the main text (Figure 2.3). In particular, the mixed codes show a faster decay of mean squared-error as SNR increases, but the full-order code decays to a larger mean squared-error baseline than either of the other codes. This baseline mean squared-error is entirely due to the input noise, and cannot be reduced by increase of the code SNR. As in the case without input noise, increasing the response field size (σ_{rf}) of the neurons in the code is likely to increase performance and correct some of the errors made due to input noise. The degree to which this changes performance will be explored in future research.

2.6.6 The rate-distortion bound and mutual information calculation

To calculate the rate-distortion bound (RDB) for our source distribution, we use a Python implementation of the iterative Blahut-Arimoto algorithm[151, 152]. Since the optimization problem is convex, the algorithm is guaranteed to converge on the right solution, given enough iterations. To ensure an adequate number of iterations, we terminate the algorithm only when successive steps are less than 10^{-10} change in error probability magnitude.

To evaluate our codes alongside the RDB, we must calculate the mutual information between

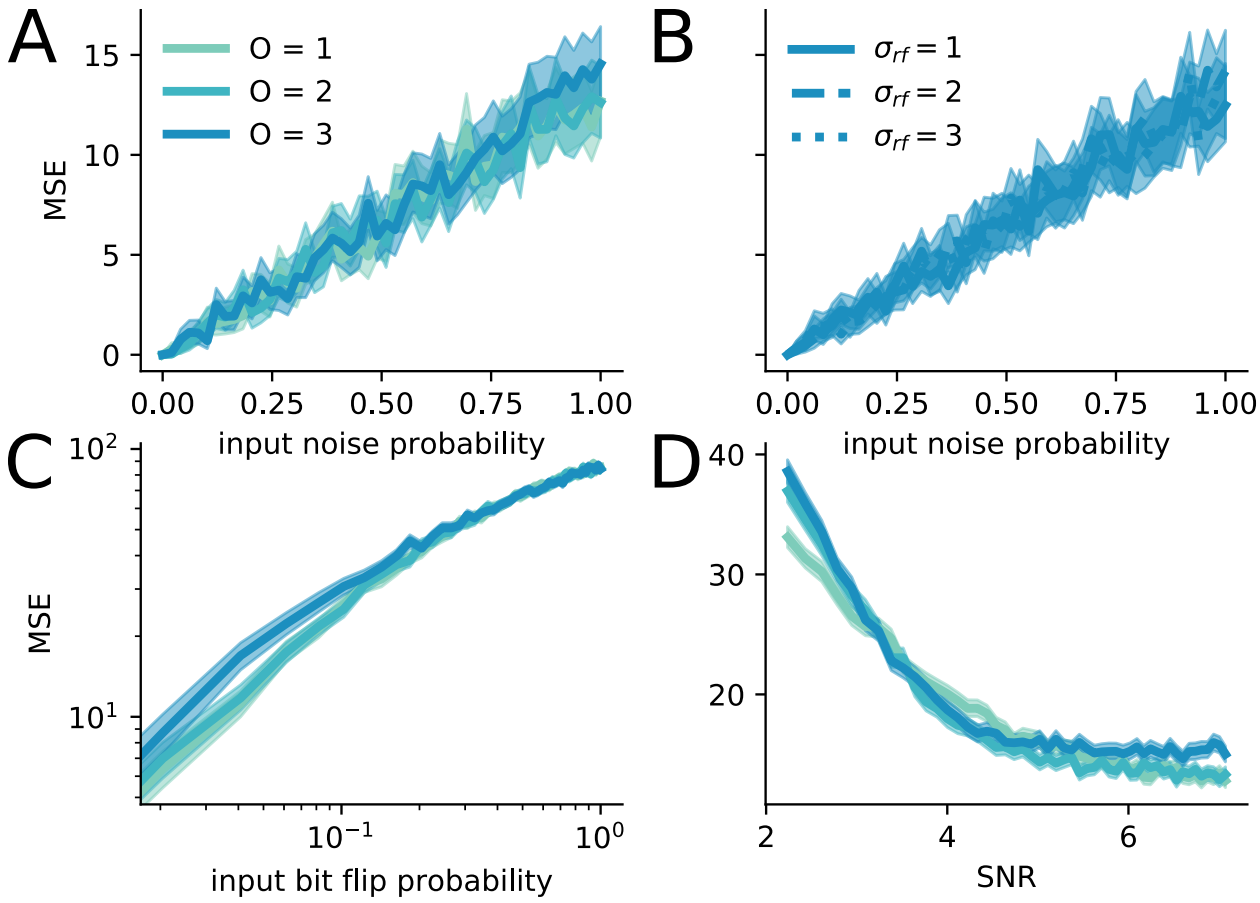

Figure 2.7: Code order does not have an effect on sensitivity to local input noise, related to Figure 2.2. For all panels, $K = 3$, $n = 10$. **A** The mean squared-error (MSE) of different codes as a function of input noise without output noise, represented as the probability of each feature taking on the value above or below its "true" value. **B** The same as **A** but for the $O = 3$ code with different RF sizes. **C** An additional simulation with non-local input noise – where bits in an input $O = 1$ code are randomly flipped with the probability given on the x-axis. The error rate of the resulting $O = 1, 2, 3$ codes with the same parameters as above is plotted. **D** A simulation with non-local input noise and output noise. The result here is similar to that without input noise in Figure 2.3, except that the $O = 3$ code has a higher error rate at high SNR due to its increased sensitivity to input noise, shown in **C**.

the stimulus distribution X and the distribution of our stimulus estimates $\hat{X}$. So,

$$I(X; \hat{X}) = H(\hat{X}) - H(\hat{X}|X)$$

where

$$
\begin{aligned}
H(Y) &= -\sum_{y \in Y} P(y) \log_2 P(y) \\
H(Y|Z) &= -\sum_{z \in Z} P(z) \sum_{y \in Y} P(y|z) \log_2 P(y|z) \\
&= \sum_{z \in Z} P(z) H(Y|Z = z)
\end{aligned}
$$

To compute these quantities, we rely the observation that $P(X) = P(\hat{X})$. That is, both distributions are uniform, with $P(\hat{x}) = P(x) = \frac{1}{n^K}$. This can be seen from the fact that none of our codewords have more (or fewer) neighbors at any given distance than any of our other codewords (see statement 4).

Using this,

$$
\begin{aligned}
I(X; \hat{X}) &= H(\hat{X}) - H(\hat{X}|X) \\
&= H(X) - H(\hat{X}|X) \\
&= K \log_2 n - \sum_{x \in X} P(x) H(\hat{X}|X = x)
\end{aligned}
$$

Since $P(x) = \frac{1}{n^K}$ and $P(\hat{X}|X = x)$ has the same entropy for all x, following from the observation above, it is enough to estimate

$$
I(X; \hat{X}) = K \log_2 n - H(\hat{X}|X = x)
$$

for a particular x. We do this via numerical simulations (see *Full channel details* in *Methods* for details).

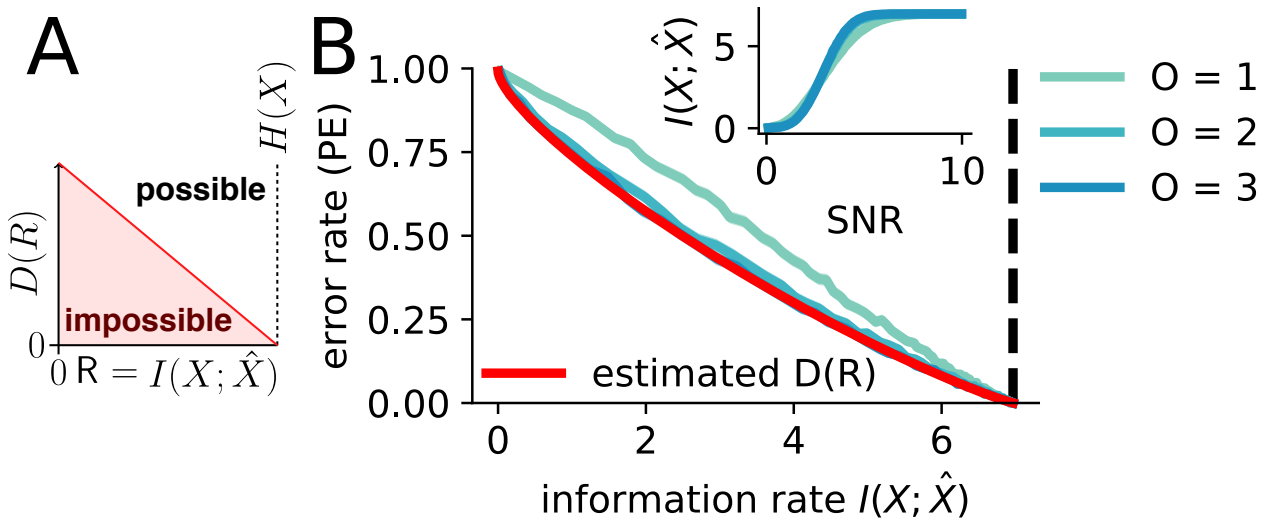

Figure 2.8: The mixed codes come close to or achieve the rate-distortion bound while the pure code does not, related to Figure 2.2. **A** A schematic of the rate-distortion bound. The bound is a function on the information rate-error rate plane dividing a region of possible codes from a region of impossible codes. The bound depends only on the stimulus distribution and distortion type, it does not depend on any code properties. Thus, we evaluate codes relative to the bound. If a code achieves the bound, that means it achieves the most efficient possible mapping from stimulus information to distortion – i.e., it uses the fewest possible bits to achieve a particular error rate. The rate-distortion bound goes to zero as $I(X;\hat{X})$ approaches $H(X)$ since the mutual information between the stimulus and its estimate cannot exceed the entropy of the stimulus. **B** For $K = 3$, $n = 5$ and a uniform probability distribution over the stimuli, we evaluated codes with different levels of mixing relative to the rate-distortion bound (red). We show that the two mixed codes $O = 2$ and $O = 3$ achieve or come close to achieving the rate-distortion bound, while the pure code does not. (inset) The transformation from SNR to $I(X;\hat{X})$ for each of the codes is fairly similar, though the mixed codes are slightly less efficient at low SNR and slightly more efficient at high SNR.

2.6.7 Representation energy required to reach a .1% error rate

We also compared codes on the basis of how much representation energy they required reach a .1% error rate given a fixed noise variance. These results are given in Figure 2.9, for noise variance $\sigma^2 = 10$.

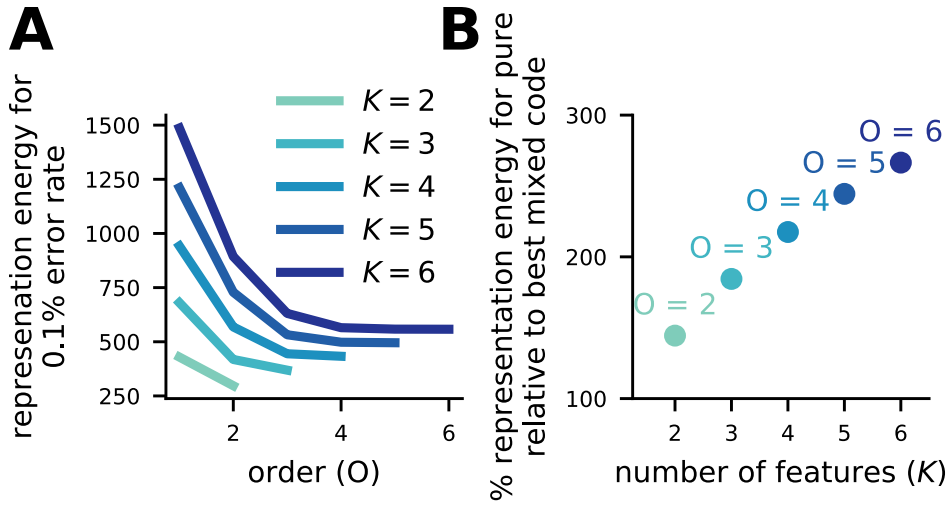

Figure 2.9: Mixed codes require less representation energy to achieve the same error rates as pure codes, related to Figure 2.2. For both plots, $n = 5$ and the noise variance $\sigma^2 = 10$. **A** The amount of representation energy required to reach a 1% error rate for codes of all orders given various numbers of features K. The code requiring the least energy is always the $O = K$ or $O = K - 1$ code. **B** The percent more representation energy required by the pure code to reach a 1% error rate compared to the optimal mixed code. The order of the optimal mixed code is indicated by the text above each marker.

2.6.8 Additional results on response fields

Generalizing our current framework to allow flexibly sized response fields (RFs) requires only a reformulation of the indicator function. Instead of performing an equality operation, it should instead perform a set membership operation, as

$$[i \in J] = \begin{cases} 0 & i \notin J \\ 1 & i \in J \end{cases}$$

where the set J is, in this case, a contiguous sequence of feature values of length σ_{rf}. Following this, for our main results, $\sigma_{\text{rf}} = 1$. Now, we explore how choosing $\sigma_{\text{rf}} > 1$ changes our results.

Effects on minimum distance, representation energy, and population size

Population size and representation energy change with RF size to ensure that full coverage of the stimulus set is maintained. To achieve this, we arrange the code dimensions in a series of σ_{rf} overlapping lattices, where each lattice has non-overlapping RFs in a grid pattern. This strategy is not guaranteed to be the most efficient tiling of the space, but it is simple to implement and analyze – and it approximately meets the theoretical estimate of the dimensionality of the most efficient tiling[133]. This RF tiling does, however, cause the stimuli on the edge of stimulus space to behave differently from the stimuli near the center. In particular, stimuli within the RF-width of the maximum or minimum feature value for one or more features will have fewer neighbors than other stimuli and will therefore have lower error probabilities than more central stimuli. Thus, for our simulations with $\sigma_{\text{rf}} > 1$, the fact that we sample stimuli uniformly rather than with some other distribution does have a mild effect on our results. However, since the number of edge stimuli is a feature of the stimulus space, not the code, the proportion of edge to non-edge stimuli is the same across codes, thus different codes do not benefit more from the sampling of additional edge stimuli.

The increase of σ_{rf} has the following effects on our three principle code metrics.

Dimensionality:

$$D_O = \binom{K}{O} \sigma_{\text{rf}} \left(\frac{n}{\sigma_{\text{rf}}} + 1 \right)^O$$

Power:

$$P_O = \binom{K}{O} \sigma_{\text{rf}}$$

Minimum distance:

$$\Delta_O = \left[2\binom{K-1}{O-1}\right]^{\frac{1}{2}}$$

Note that minimum distance is not affected.

The optimal σ_{rf} for a given total energy

For a fixed K, O, n, and E, we want to find the σ_{rf} that maximizes minimum distance. For $E = \epsilon V + D_O$, and using $\delta(K, O, \sigma_{\text{rf}}, V)$ as an expression for minimum distance after application of β to produce a code with power V, we can write the problem as:

$$\begin{aligned} L &= \delta\left(K, O, \sigma_{\text{rf}}, \frac{E - D_O}{\epsilon}\right)^2 \\ &= \frac{2O}{K\epsilon}\left(\frac{E - D_O}{\sigma_{\text{rf}}}\right) \\ &= \frac{2O}{K\epsilon}\left[\frac{E}{\sigma_{\text{rf}}} - \binom{K}{O}\left(\frac{n}{\sigma_{\text{rf}}} + 1\right)^O\right] \end{aligned}$$

and now, to find the maximum, we will take the derivative $\frac{\partial L}{\partial \sigma_{\text{rf}}}$,

$$\begin{aligned} \frac{\partial L}{\partial \sigma_{\text{rf}}} &= \frac{2O}{K\epsilon}\frac{\partial L}{\partial \sigma_{\text{rf}}}\left[\frac{E}{\sigma_{\text{rf}}} - \binom{K}{O}\left(\frac{n}{\sigma_{\text{rf}}} + 1\right)^O\right] \\ &= \frac{2O}{K\epsilon}\left[-\frac{E}{\sigma_{\text{rf}}^2} + \binom{K}{O}O\left(\frac{n}{\sigma_{\text{rf}}} + 1\right)^{O-1}\frac{n}{\sigma_{\text{rf}}^2}\right] \end{aligned}$$

and now setting the LHS to zero,

$$\begin{aligned}
\frac{\partial L}{\partial \sigma_{\text{rf}}} = 0 &= \frac{2O}{K\epsilon}\left[-\frac{E}{\sigma_{\text{rf}}^2} + \binom{K}{O} O \left(\frac{n}{\sigma_{\text{rf}}} + 1\right)^{O-1} \frac{n}{\sigma_{\text{rf}}^2}\right] \\
E &= \binom{K}{O} O \left(\frac{n}{\sigma_{\text{rf}}} + 1\right)^{O-1} n \\
\frac{E}{\binom{K}{O} O n} &= \left(\frac{n}{\sigma_{\text{rf}}} + 1\right)^{O-1} \\
\left(\frac{E}{\binom{K}{O} O n}\right)^{\frac{1}{O-1}} &= \frac{n}{\sigma_{\text{rf}}} + 1 \\
\sigma_{\text{rf,opt}} &= n\left[\left[\frac{E}{O n \binom{K}{O}}\right]^{\frac{1}{O-1}} - 1\right]^{-1}
\end{aligned}$$

See Figure 2.10F for a plot of this function. This formalization does ignore benefits of $\sigma_{\text{rf,opt}} > 1$ for reducing the number of nearest neighbors of high order codes.

Effects on error distribution

Increasing RF size has the effect of pulling the distribution of squared-error distortion more concentrated toward zero, while increasing the overall probability of an error (see Figure 2.3D). The increase in overall probability of an error for fixed SNR can be understood by the expression for code power given above, where an increase in RF size increases the power consumption of the code without producing a change in minimum distance.

However, increasing RF size does produce a change in the number of codewords at minimum distance and at succeeding distances. To see this, we can focus on the $O = K$ case: with $\sigma_{\text{rf}} = 1$, we know that all other codewords are nearest neighbors to a given codeword (Eq. 2.13) because only one dimension is active for each codeword. If, instead, we have $\sigma_{\text{rf}} = 2$, we know that each RF has a volume of σ_{rf}^K feature values, but their intersection must be of size 1. Thus, either active RF can be changed to $\sigma_{\text{rf}}^K - 1$ different RFs to still form a valid

codeword. Thus, the number of nearest neighbors is $2\left(2^K - 1\right)$. With $\sigma_{\text{rf}} = 2$, all stimuli except the nearest neighbors will be at the same, further distance.

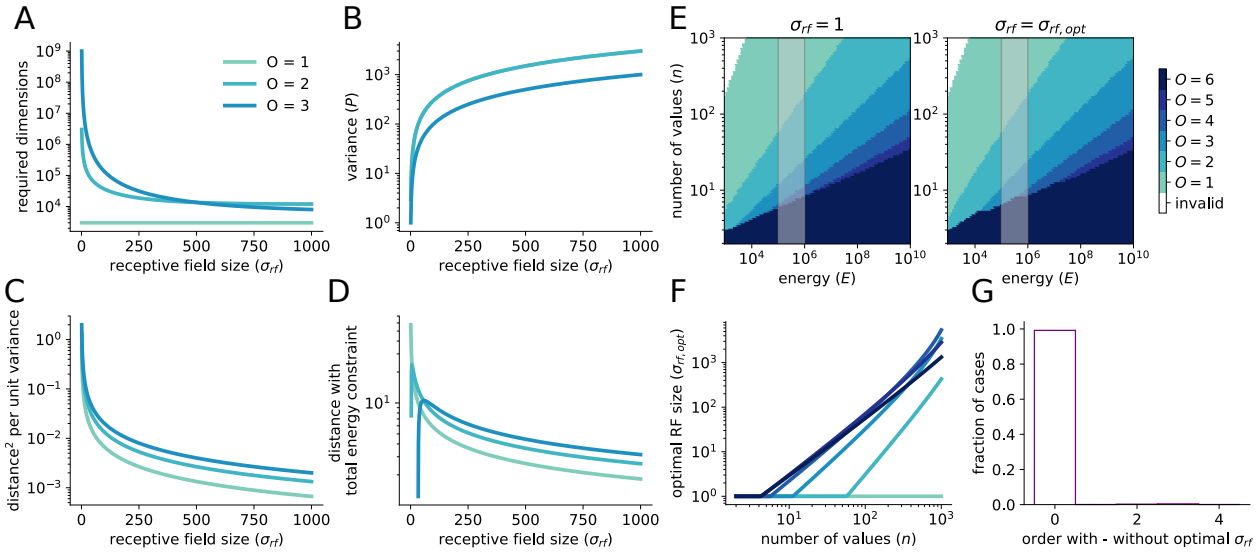

Figure 2.10: Changing response field (RF) size changes code properties, related to Figure 2.3. **A** The number of dimensions required to implement the code decreases by several orders of magnitude. **B** The power of the code increases by several orders of magnitude. **C** The tradeoff between minimum distance and code power remains constant if all codes are given the same RF size. **D** The RF size maximizing minimum distance under the total energy constraint differs between codes. **E** The code providing the highest minimum distance with $\sigma_{\text{rf}} = 1$ (left) and $\sigma_{\text{rf}} = \sigma_{\text{rf,opt}}$ (right) as computed in Eq (S.2). They are only marginally different. **F** The optimal RF size for codes of different orders given features with different numbers of possible values. **G** Histogram of the differences in code order giving the highest distance from **E**.

2.6.9 Error-reduction by mixed selectivity in the continuous case

Here, we adopt continuous stimulus features and RFs to test how well the benefits of mixed codes generalize to the continuous case (also see [65] for a deeper investigation of the continuous case). In particular, with stimuli $x \in X$ composed of K features, $x_i \sim U\left(0, n_i\right)$. Instead of the flat, discrete RFs defined in *Additional results on response fields* in *Supplement*, we

use Gaussian RFs,

$$r(x|\sigma_{\mathrm{w}}, c) = \exp\left(-\frac{\sum_i^{D_O}(x_i - c_i)^2}{2\sigma_{\mathrm{w}}^2}\right)$$

which are then scaled by the amplifying transform β as described in *Linear transform (β)* in *Methods*. The rest of the channel is identical to the channel described previously, including the additive noise. RFs are tiled in the same way, though now their width σ_{w} is independent of σ_{rf}, which dictates their tiling – as in *Additional results on response fields* in *Supplement*.

Our simulations show similar results to the discrete case (Figure 2.11), with higher order codes yielding lower MSE across all of the SNRs we investigated. Thus, the broad advantage of mixed codes apply in the continuous case as well. However, increasing RF size produces higher MSE, which is the opposite of our results in the discrete case. Future work is needed to discover why this is, and in what other ways the continuous case differs from the discrete case.

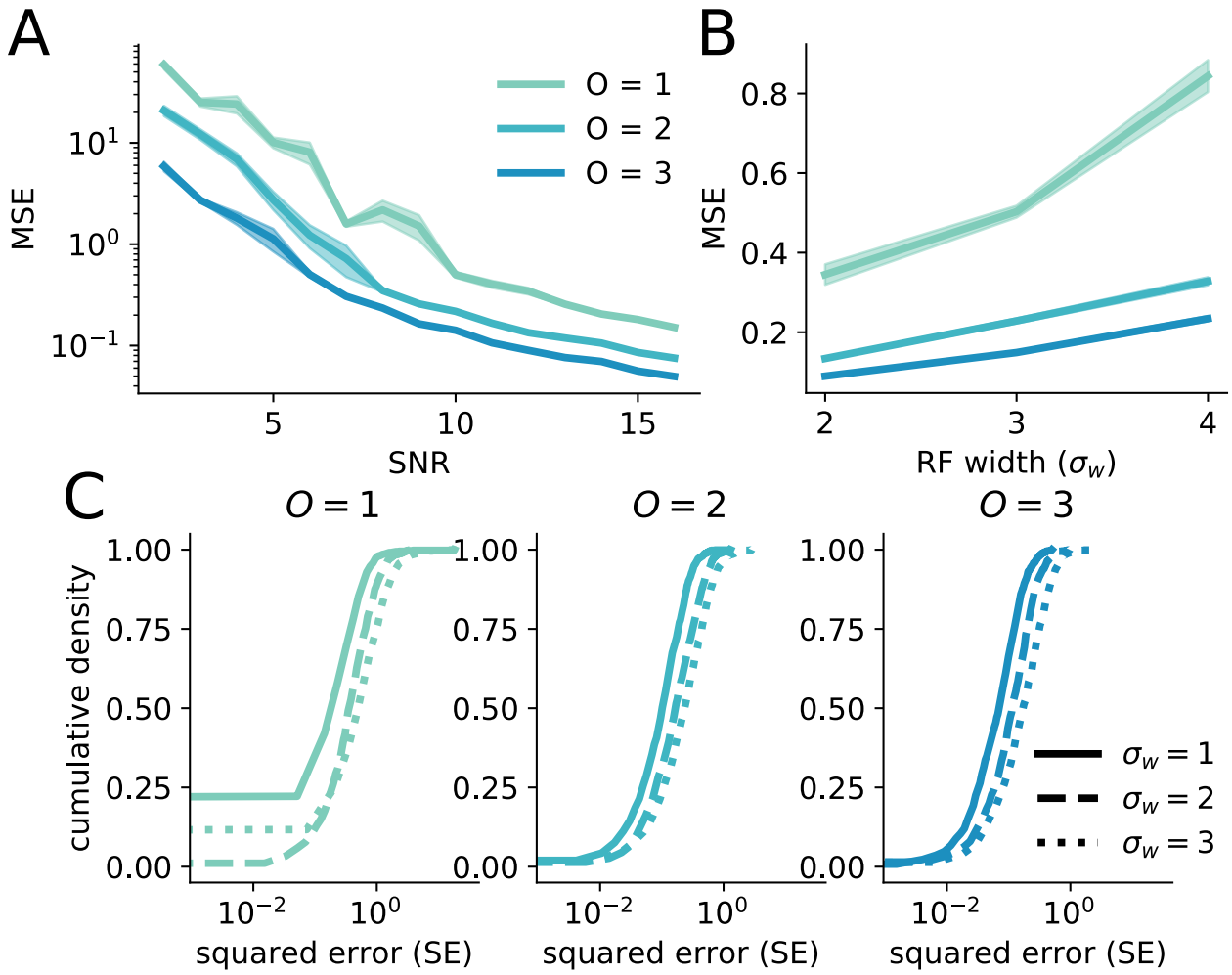

Figure 2.11: The benefits of mixed codes broadly generalize to continuous stimuli and RFs, related to Figure 2.3. **A** The MSE of codes of all orders with $K = 3$. The higher-order codes provide better performance than the lower-order codes. **B** MSE increases with RF size, which is contrary to the result in the discrete case (Figure 2.3d). **C** The cumulative distribution function of squared error for the three codes and for three different RF sizes.

CHAPTER 3

SOLUTIONS TO THE ASSIGNMENT PROBLEM BALANCE TRADE-OFFS BETWEEN LOCAL AND CATASTROPHIC ERRORS

W. Jeffrey Johnston and David J. Freedman

Abstract

An individual observing a barking dog and purring cat together in a field has distinct pairs of representations of the two animals in their visual and auditory systems. Without prior knowledge, how does the observer infer that the dog barks and the cat purrs? This binding of disparate representations is called the assignment problem, and it must be solved to integrate distinct representations across but also within sensory modalities. Here, we identify and analyze a solution to the assignment problem: the representation of one or more common stimulus features in pairs of relevant brain regions – for example, estimates of the spatial position of both the cat and the dog represented in both the visual and auditory systems. We characterize the reliability of this solution as well as how that reliability depends on different features of the stimulus set (e.g., the size of the set and the complexity of the stimuli) and the details of the split representation (e.g., the precision of each stimulus representation and the amount of overlapping information). Next, we consider a representation with limited capacity and determine how the assignment error rate and precision of stimulus representations depend on that capacity. This constraint reveals two trade-offs between the precision of individual stimulus representations and the frequency of catastrophic assignment errors. These trade-offs balance redundancy, which reduces the frequency of assignment errors, and efficiency, which increases the precision of stimulus representations. Further, we show that assignment errors reported in humans are broadly consistent

with those predicted by our model, as well as make further testable predictions. Overall, this work offers insight into an important computational problem that arises from distributed neural representations.

Keywords: multisensory integration, representation assignment, information theory

3.1 Introduction

Humans and other animals successfully behave in highly complex environments, composed of multiple objects each of which might produce representations in multiple sensory systems. Coherent behavior in these environments requires extensive integration of all these disparate forms of information about the environment. For instance, a hawk bringing food to her hatchlings might need to attend to both the visual appearance of each hatchling and their vocalizations to infer which of them is most in need of food. While navigating these cluttered multisensory environments often appears effortless for hawks and other species, it requires two highly non-trivial computations: Object segmentation and representation assignment. For object segmentation, the continuous sensory world must be segmented into distinct objects – that is, each hatchling has to be recognized as a distinct entity, separated from both the other hatchlings and other background objects, such as the nest. The primitive rules that humans and other animals use to segment objects have been studied extensively as part of gestalt psychology[153] and related subfields[154]. In addition, the neural mechanisms underlying object segmentation are beginning to be understood, with the discovery of neurons that appear to signal which nearby object a particular edge belongs to, termed border ownership cells, in the primate visual system[155, 156]. However, even once object representations are segmented within a single brain region, that set of object representations needs to be integrated with distinct sets of representations of those same objects that arise in different brain regions. We refer to this process as representation assignment. In particular,

in the example above, the representations of the chicks from the visual system need to be integrated with the representations of the chicks from the auditory system, as both sensory systems provide information that is necessary to guide coherent behavior. The integration of distinct sets of representations of the same objects is called the assignment problem[89].

A reliable solution to the assignment problem is crucial to coherent behavior, especially as sensory information is known to be widely distributed in the brain. In particular, errors in assignment lead to incorrect conjunctions of object features, producing representations – such as a barking cat and meowing dog, see Figure 4.1a – that are likely to be particularly catastrophic for behavior. The integration of these distinct, parallel representation of the world has been previously studied in two principle ways. First, the integration of distinct features within the visual system has been studied in the context of feature integration theory[90, 91]. In feature integration theory, spatial attention is used exclusively to bind features together due to their spatial proximity. Spatial attention is deployed to different locations in sequence and only a single object is bound at a time. However, in experiments, humans have been shown to make illusory conjunctions of stimulus features[157, 158], where they associate the features of one object with a different object. These illusory conjunctions are a form of assignment error. While feature integration theory provides a qualitative description of the assignment process and of assignment errors, it does not provide a mechanistic explanation of or quantitative predictions for assignment errors.

Second, representation assignment has also been studied in the context of multisensory integration[83, 86–89]. In this literature, both experiment and theory have primarily focused on the integration of multisensory representations of a single stimulus. This work has demonstrated that shared information is crucial to reliable integration. In particular, integration of auditory and visual information appears to rely on the shared representation of azimuthal position derived from both sensory systems[88]. That is, whether or not an auditory and

visual stimulus are integrated depends on depends on the mean squared-error (MSE) of the representations of azimuthal position in both sensory systems[88]. We refer to the MSE of these individual representations as the local distortion.

Here, we extend this analysis to sets of multiple objects and develop a general framework for solutions to the assignment problem that rely on multiple common feature representations, such as estimates of both azimuthal position and elevation. Then, we analyze the reliability and efficiency of this solution to the assignment problem. We show that increasing the number of commonly represented features drives the assignment error rate down, and increases the level of redundancy between the two representations. When fixing the available metabolic resources, this increase in redundancy leads to an increase in the local distortion (MSE) of individual feature representations. We illustrate a similar trade-off for representations of the common stimulus features with low local distortion in one region and high local distortion in the other, even when the two sources of information are optimally combined. We show that both of these trade-offs can be leveraged to minimize the mean squared-error (MSE) of the integrated object representations. Finally, we link our model framework to experimental data from the human working memory literature, and show that there is close fit to the data for three quantitative predictions made by our framework. This work demonstrates a general solution to the assignment problem, which is an important part of the binding problem. This solution implies a general trade-off between catastrophic assignment errors and the magnitude of local decoding errors, that is navigated by changes to the level of redundancy between distinct representations of the same objects.

3.2 Results

In the brain, multiple distinct sensory systems and brain regions give rise to multiple distinct sets of representations of the same objects. To guide reliable behavior, these distinct sets

of representations must be integrated. That is, the brain must infer the correct set of multi- and unisensory objects (i.e., hidden causes) that give rise to its distributed sets of sensory representations. Redundant information between sensory systems and brain regions is necessary to correctly infer the underlying objects without making assignment errors, in which the features of one stimulus are integrated with the features of a different stimulus. These assignment errors are likely to be particularly catastrophic for behavior, because they produce representations of, for instance, a barking cat and a purring dog. Here, we show how redundancy between sensory systems and brain regions mediates a trade-off between catastrophic assignment errors and the magnitude of local errors.

In our framework, an object is represented by a single point in a K-dimensional space. The representation of that object is split into two subspaces – i.e., brain regions R_X and R_Y – which each represent a subset of the K features of the object. As an example, we consider R_X to be an auditory brain region and R_Y to be a visual brain region (Figure 4.1a). Next, we take there to be both a dog and a cat present in the world – that is, the set of objects (or stimuli) in the world S consists of a dog and a cat. Then, $\hat{X}$ is a set of two representations, one of each of the animals' vocalizations; similarly, $\hat{Y}$ is a set of two representations, one of each of the animals' visual features. Then, we want to integrate the auditory and visual features of the dog and cat – that is, recover an estimate of the full K-dimensional objects from the incomplete representations in R_X and R_Y. In our example, there are two possible one-to-one mappings between the auditory and visual representations. One mapping produces the correct assignment and links the dog's visual features to its vocalization – and similarly for the cat (Figure 4.1a, bottom left) – and the other produces the incorrect assignment and links the dog's visual features to the cat's vocalization – and similarly for the other representations (Figure 4.1a, bottom right). Here, we ask how representations can be designed so that the correct mapping is reliably chosen and assignment errors are avoided.

We show that the most likely mapping between the two sets of representations depends on redundancy between the representations in R_X and R_Y. If the two sets of representations are completely independent of each other, then the best possible integration strategy is to select a mapping randomly. Alternatively, redundancy between the two sets of representations can be produced by common representations of the same object features. As in our example above, both the auditory and visual systems develop a representation of the azimuthal position. This redundant representation can be used to solve the assignment problem. In general, for C commonly represented stimulus features, the two sets of representations are in a common C dimensional space. Without any neural noise, the two sets of representations would be perfectly overlapping in the common C-dimensional space, and the correct one-to-one mapping would integrate the overlapping object representations with each other (Figure 4.1b). With neural noise that produces noisy feature representations, such that each feature represented in R_X is decoded with local distortion (or MSE) D_X, we show that the most likely one-to-one mapping between those two sets of representations is the one that minimizes the sum of squared distances between each integrated pair (Figure 4.1c). Further, we show that assignment errors occur precisely when the representation of one object crosses over the representation of another object in one region but not the other (Figure 4.1c, red lines). For two objects at a fixed distance from each other in the shared space δ, this event has probability approximated by

$$F(\delta) \approx Q\left(\frac{-\delta}{\sqrt{2D_X}}\right) + Q\left(\frac{-\delta}{\sqrt{2D_Y}}\right)$$

where $Q(x)$ is the cumulative density function for the standard normal distribution (Figure 4.1d, left) – for the full expression and derivation, see *Probability of assignment errors* in *Methods*.

From this, we can derive the overall probability of an assignment error. In particular, the

overall probability of an assignment error also relies on the probability that two objects are at distance δ from each other in the common space $p_C(\delta)$ (Figure 4.1d, right for $C = 1$) and on the number of stimuli. Incorporating these, we show that the overall probability of an assignment error is upper bounded by,

$$AE_C \leq \binom{N}{2} \int d\delta \, p_C(\delta) \, F(\delta) \tag{3.1}$$

where N is the number of stimuli. Further, we show that this expression can be approximated in closed form for $C = 1$,

$$AE_1 \approx \binom{N}{2} \frac{2\sqrt{D_X + D_Y}}{s\sqrt{\pi}} \tag{3.2}$$

where s is the size of the feature space for the commonly represented feature. This approximation (dashed lines, Figure 4.1e) closely matches the empirical assignment error rate (solid lines) across different numbers of stimuli (different colors; see *Assignment error rate approximation for* $C = 1$ in *Methods* for a full derivation of this expression). Now, using this formalization, we characterize how the assignment error rate changes with additional commonly represented features $C > 1$ and asymmetric estimator variance for those common features – that is, $D_X \neq D_Y$.

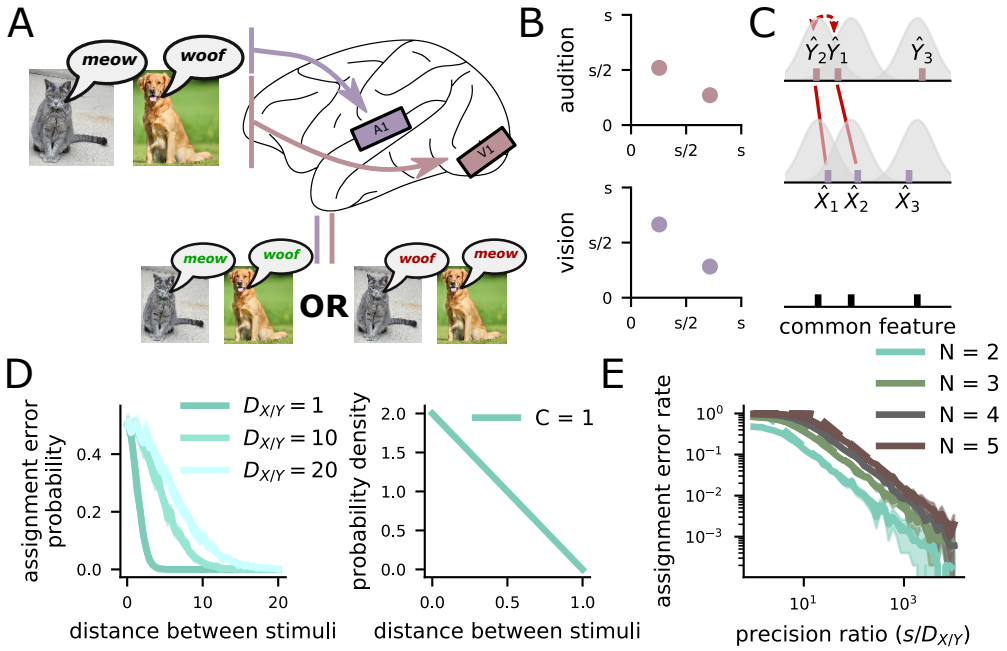

Figure 3.1: The assignment problem arises from distributed representations of the world, and can be solved by redundant representations. **A** A schematic of the assignment problem. The brain receives both visual and auditory information about a dog and a cat, this information is initially separated in the brain. When combining the two sets of representations (auditory and visual), there are two possible integrations, one that correctly reconstructs a barking dog and meowing cat (left) and the other that incorrectly constructs a barking cat and meowing dog (right). The right is an example of an assignment error. **B** The assignment problem can be solved by the representation of a common stimulus feature in both brain regions. In our example, the auditory (top) and visual (bottom) representations can be integrated through a shared representation of azimuthal position (the shared x-axis). **C** Due to the variability in neural systems, the estimate of the common feature value will have some distribution within each region (top and middle). We assume that distribution is Gaussian and centered on the true value for each of three different stimuli, with variance that we refer to as D_X and D_Y for features represented in R_X and R_Y, respectively. In these conditions, assignment errors will occur when the estimate of the common feature value for two stimuli cross over each other in one region, but not the other (middle, red arrow). **D** (left) The probability that this crossing over occurs is high for nearby stimuli and low for distant stimuli, and increasing estimator variance makes assignment errors more likely at all distances. (right) For stimuli that are uniformly distributed in the full feature space, the distance between pairs of stimuli follows a triangular distribution with one commonly represented feature ($C = 1$). **E** The overall assignment error rate is the product of the two functions on the left – the assignment error probability at each distance weighted by the probability that there is a pair of stimuli at that distance. We develop an closed form approximation of this product (dashed lines) which is well matched by simulation results (solid lines) for different numbers of stimuli.

3.2.1 The assignment error rate depends on feature distortion and overlap

While distinct sensory systems tend to have fixed amounts of overlapping information, such as a common estimate of azimuthal position across the auditory and visual systems (which has been shown to be crucial for single stimulus integration[88]), within a single sensory system information is distributed across multiple brain regions, and those brain regions can represent variable amounts of overlapping information about the stimuli. In our framework, we show how increasing the number of commonly represented stimulus features across the two brain regions C changes the assignment error rate. Further, the amount of overlapping information between two brain regions can also be altered by changing the estimator variance of common feature representations in both regions – that is, D_X and D_Y.

First, we show how the assignment error rate depends on the number of overlapping stimulus features C for constant estimator variance. Since estimator variance is kept constant, the changes in assignment error rate that result from changes to the number of commonly represented stimulus features are due only to the increased dimensionality of the commonly represented feature space. Adding additional dimensions to the commonly represented feature space can only increase the distance between pairs of points in that space (Figure 4.2a) – and will only fail to increase the distance if the stimuli are likely to have the same value in the new dimension. This change is captured by changes in the $p_C(\delta)$ term in Eq. 3.1. Increasing the number of commonly represented features moves the probability mass of this distribution away from zero (Figure 4.2b), and increases the mean to such an extent that even one additional common stimulus feature reduces the likelihood of an assignment error between stimuli at that mean distance by several orders of magnitude (Figure 4.2c) – this effect holds for Gaussian as well as uniform stimuli, and, as remarked above, would hold to some degree for stimuli with any distribution that does not have all of its probability mass at a single value. Because the function $AE(\delta)$ is strictly decreasing with δ, having larger

distances δ be more probable and, thus, more heavily weighted in Eq. 3.1 by $p_C(\delta)$ will lead to a decrease in the overall assignment error rate. For the same estimator variance, including two rather than one commonly represented feature can decrease the assignment error rate by multiple orders of magnitude (Figure 4.2d).

One way for quantifying the information available for use in solving the assignment problem is through the redundancy between the estimated representations $\hat{X}$ and $\hat{Y}$, which can be written as the mutual information $R = I(\hat{X};\hat{Y})$ between these two sets of random variables. The redundancy captures how much one can guess about the values of $\hat{Y}$ given only the values of $\hat{X}$. If there are no commonly represented features $C = 0$ and all of the K features are independent of each other, then the redundancy is zero and the assignment problem cannot be reliably solved when there is more than a single stimulus. If there are commonly represented features $C > 0$, then the redundancy is also greater than zero. In fact, the redundancy is proportional to the number of overlapping features. So, for constant D_X and D_Y, increasing C will strictly increase the amount of redundancy between the two representations (Figure 4.2e). The redundancy can also be changed by adjusting D_X and D_Y: Increasing (decreasing) both will decrease (increase) the redundancy, but increasing one and decreasing the other will also decrease the redundancy in certain conditions.

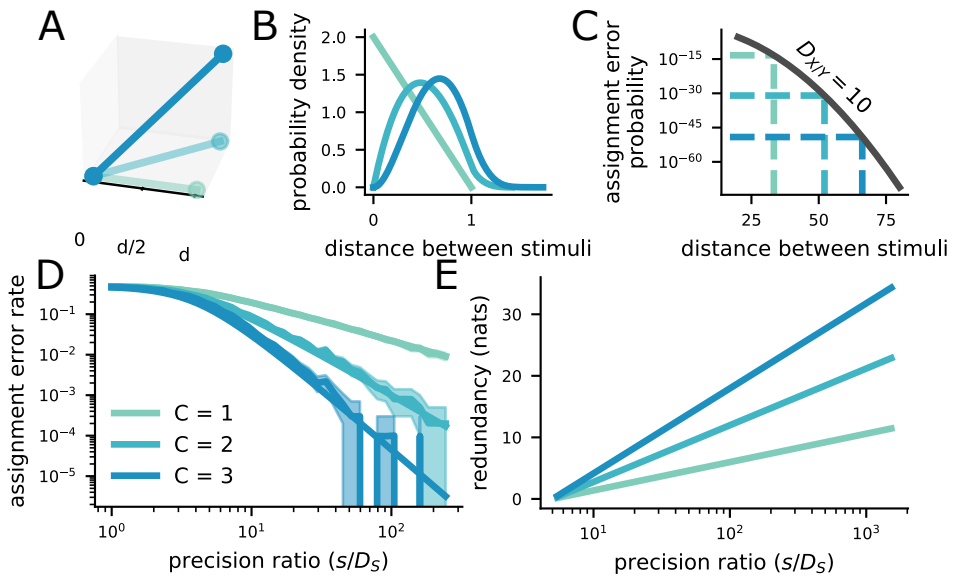

Figure 3.2: Increasing the number of commonly represented features decreases the assignment error rate, but increases the level of redundancy between the representations. The color legend is the same throughout the plot, and provided in **D**. **A** Schematic showing that the distance between two points increases with the dimensionality of the space that they are in. **B** The distribution of distances between two uniformly distributed points in a space of one, two, and three dimensions. The distribution shifts to the right (toward larger distances) as the dimensionality of the space increases. This is true for points with any distribution that has non-zero variance. **C** At the average distance between two points (dashed lines), the probability of an assignment error decreases by orders of magnitude as the dimensionality of the commonly represented space increases, without changing the estimator variance for the representations $D_{XY} = 10$. **D** The overall assignment error rate also decreases by orders of magnitude as the dimensionality of the commonly represented feature space increases, while holding D_{XY} constant. The difference becomes even larger as the precision ratio increases. **E** The redundancy between representations $\hat{X}$ and $\hat{Y}$ also increases as the dimensionality of the commonly represented feature space increases – again, the difference is increased at larger precision ratios. Thus, the assignment error rate is driven down at the cost of additional redundancy.

In general, the representations of the same stimulus feature may not be encoded with the same fidelity in each region. For instance, estimates of azimuthal position from the visual system are thought to be much more accurate than those from the auditory system[88]. Further, the relative accuracy of the azimuthal position and elevation of sound shifts across sensory field[159]. Next, we show how these asymmetric feature representations affect the assignment error rate. From Eq. 3.1, we see that the assignment error rate only depends on the sum of the estimator variances across the two regions. As that sum increases, the assignment error rate will also increase. We assume that the two sources of stimulus information are optimally combined by the downstream brain region, such that the estimator variance of a commonly represented feature D_S is given by the following relationship,

$$D_S = \frac{D_X D_Y}{D_X + D_Y}$$

where D_X and D_Y are, as before, the estimator variance of the representation of that feature in regions X and Y, respectively (Figure 4.3a). Using this equation for the integrated estimator variance, we can rewrite D_X and D_Y in terms of the desired integrated estimator variance D_S and the representation asymmetry ΔD,

$$D_X = \frac{2D_S}{1 - \Delta D} \tag{3.3}$$

$$D_Y = \frac{2D_S}{1 + \Delta D} \tag{3.4}$$

where, without loss of generality, we have assumed $D_X \geq D_Y$ (Figure 4.3b) and the representation asymmetry ΔD is constrained to be on the interval $[0, 1)$. When $\Delta D = 0$, the feature representations are perfectly symmetric, as before, and $D_X = D_Y = 2D_S$; when $\Delta D \to 1$, then $D_X \to D_S$ and $D_Y \to \infty$, or vice versa. A fully asymmetric representation provides no redundancy and thus cannot be used to solve the assignment problem, effectively reducing the number of commonly represented features C.

Further, we know that the assignment error rate for a fixed number of commonly represented features is determined by the magnitudes of D_X and D_Y, shown in Eq. 3.1. To understand how representation asymmetry affects assignment errors in general, we can exploit features of the Gaussian cumulative distribution function to show that the changes in the assignment error rate due to changes in ΔD are lower bounded by an expression that depends only on the sum of D_X and D_Y. Thus, rearranging the sum, we can see that,

$$D_X + D_Y = \frac{4D_S}{1 - \Delta D^2} \tag{3.5}$$

which is strictly increasing with ΔD. Thus, increasing the level of asymmetry while keeping other features of the representations constant increases the lower bound on the assignment error rate. We verify that the assignment error rate itself increases by numerical evaluation of the assignment error rate at a variety of distances (Figure 4.3c). Finally, we evaluate the overall assignment error rate for different levels of representation asymmetry (Figure 4.3d) and show that it also increases as a function of representation asymmetry for all precision ratios. Next, we evaluate the redundancy between $\hat{X}$ and $\hat{Y}$ as a function of precision ratio and representation asymmetry. As expected, increasing representation asymmetry also decreases the level of redundancy between the two sets of representations (Figure 4.3e), as it makes the set of representations observed in one region less informative about the representations observed in the other.

In some cases, asymmetric feature representations are imposed by the physical limits of different sensory systems, such as with audition and vision; in others, however, asymmetric feature representations can be designed, such as is hypothesized to be the case across the canonical dorsal and ventral visual processing streams in the primate. In that case, representations of position are thought to be encoded with high fidelity in the dorsal visual stream, but with relatively low fidelity in the ventral visual stream. Yet, the assignment problem

must also be solved when integrating across those two sets of regions, so an asymmetric representation of spatial position across them may seem non-optimal. Next, we resolve this apparent contradiction by showing that, in some cases, asymmetric feature representations increase the efficiency of solutions to the assignment problem by reducing the redundancy between the two regions.

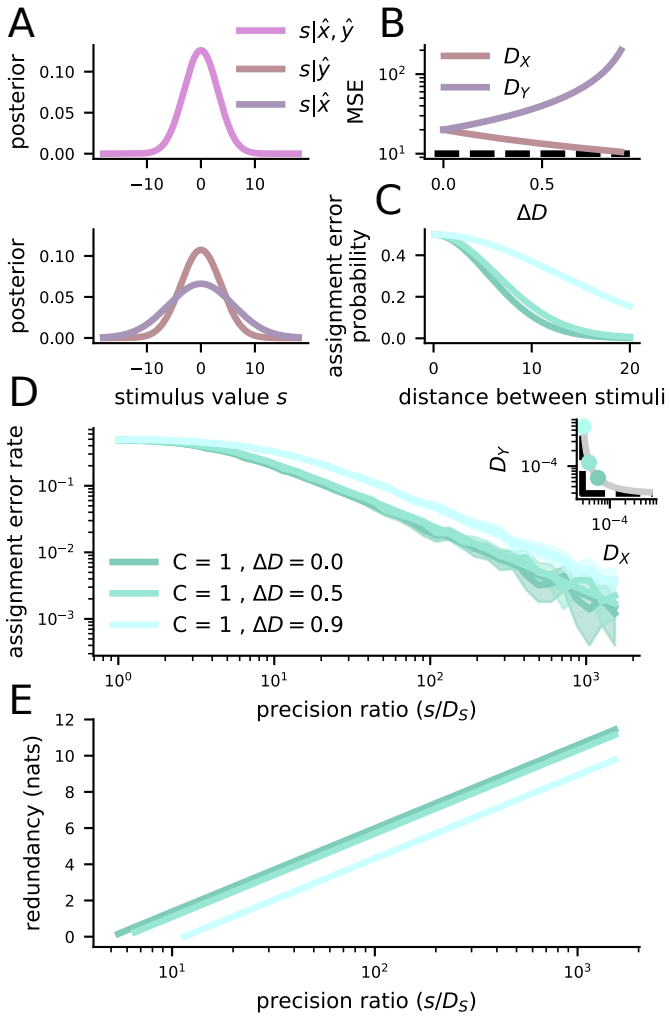

Figure 3.3: Asymmetric feature representations increase the assignment error rate, but decrease redundancy. **A** Schematic of representation integration. The stimulus estimates from R_X and R_Y are optimally combined to give rise to a stimulus estimate with variance $D_S = D_X D_Y/(D_X + D_Y)$. **B** We can keep D_S constant (dashed line) and define D_X and D_Y in terms of D_S and the representation asymmetry ΔD, which is 0 when $D_X = D_Y$ and goes to 1 when $D_X \to D_S$ and $D_Y \to \infty$, or vice versa. **C** Increasing representation asymmetry ΔD increases the assignment error rate for pairs of stimuli at all distances. **D** The overall assignment error rate increases with increases in representation asymmetry across all precision ratios. **E** The redundancy between $\hat{X}$ and $\hat{Y}$ decreases with increases in representation asymmetry across all precision ratios.

3.2.2 Limited resources constrains both the assignment error rate and feature distortion

So far, our analysis has focused on describing how the assignment error rate depends on the number of commonly represented features across two brain regions X and Y as well as on the variance of an estimator for each of those features in each of those brain regions. These results allow the analysis of existing systems, and demonstrate how the brain can be designed to avoid assignment errors. Here, we extend our analysis to include a constraint on the total amount of resources available to the brain, and show how this leads to a trade-off between the magnitude of local errors (represented by estimator variance) and the catastrophic, non-local assignment errors that we have been focusing on so far.

As our constraint, we take the total amount of Shannon information that can be allocated across all of the stimulus representations across both regions. The Shannon information has an immediate connection to estimator variance via the rate-distortion bound – further, it ties naturally to the quantification of redundancy between representations that we have already used to characterize the effect of commonly represented features and representation asymmetry. Further, neural codes have long been hypothesized to maximize Shannon information, in the literature on efficient coding[46, 47], and the Shannon information provided by neural receptive field codes in general can be easily evaluated due to an established relationship between Fisher and Shannon information in certain conditions[114] – but see [115]. In our analysis, we use the rate-distortion bound for both Gaussian and uniformly-distributed stimulus sets to characterize the minimum achievable estimator variance for a given amount of information. In particular, if we have B bits of information to use to describe a Gaussian-distributed stimulus with variance s^2, then the minimum estimator variance we could achieve is given by $s^2 \exp(-2B)$ (Figure 4.4a, see *Rate-distortion bound for a uniform object distribution* in *Methods* for a derivation). Now, if we have the same B bits of informa-

tion with which to describe a K-dimensional Gaussian-distributed stimulus, we first must decide how to allocate our B bits among the K features and, then, we can directly apply the relationship given above. If the variance of each dimension of the Gaussian is the same, then the optimal strategy is to allocate bits evenly across all the dimensions. In this case, the distortion-rate bound has the form,

$$D = s^2 \exp\left(-\frac{2B}{K}\right)$$

where s^2 is the variance of each stimulus feature and B is the number of bits available for encoding the entire K-dimensional stimulus.

In our formalization, rather than distributing bits equally, we want to distribute bits such that, in the absence of an assignment error, each feature of the stimulus will be estimated with a constant variance, D_S. Thus, we allocate a fixed number of bits to uniquely represented stimulus features (Figure 4.4b, left, red bars), but allocate a distinct number of bits to the commonly represented features, depending on the representation asymmetry across the two brain regions (Figure 4.4b, left, blue-green bars). Even given this constraint, the commonly represented features require many more bits for their representation than the uniquely represented features – and these excess bits are precisely the redundancy that we described in Figure 4.2e and Figure 4.3d.

Using the rate-distortion bound, the constraint of achieving estimator variance D_S for all features, and the optimal combination of D_X and D_Y into D_S established in the previous section, we can now write D_S in terms of all of the features of the representation system that we have so far discussed. That is, using the rate-distortion bound, we write D_S as a function of the extent of each feature s, the number of bits used in our representation B, the number of commonly represented features C, and the representation asymmetry of those

common features ΔD. This expression takes the form

$$\begin{aligned} D_S &= s^2 \exp\left(-\frac{2B - C\log\frac{1-\Delta D^2}{4}}{K+C}\right) \\ &= \left[\frac{1-\Delta D^2}{4}\right]^{\frac{C}{K+C}} s^2 \exp\left(-\frac{2B}{K+C}\right) \end{aligned} \tag{3.6}$$

from which we can see that, as we expect, increasing feature representation asymmetry ΔD will decrease the achievable estimator variance for a fixed number of bits. Thus, choosing asymmetric feature representations will improve the local fidelity of stimulus representations. Further, we can see that increasing C will increase D_S. Both of these manipulations change the redundancy between X and Y, which we can now quantify explicitly,

$$\begin{aligned} R &= I(X;Y) = H(X) - H(X|Y) \\ &= K_X \frac{1}{2}\log 2\pi s^2 - (K_X - C)\frac{1}{2}\log 2\pi s^2 - C\frac{1}{2}\log 2\pi(D_X + D_Y) \\ &= C\frac{1}{2}\log 2\pi s^2 - C\frac{1}{2}\log 2\pi(D_X + D_Y) \\ &= \frac{C}{2}\log\frac{s^2}{D_X + D_Y} \\ &= \frac{C}{2}\log\frac{s^2(1-\Delta D^2)}{4D_S^2} \end{aligned}$$

where R is the redundancy, in bits, between the representations in region X and Y. This redundancy is crucial for solving the assignment problem. The redundancy is proportional to the number of commonly represented features C – so, increasing C will produce relatively large increases in redundancy, and, as we have seen, can be expected to effectively reduce assignment errors. Further, increasing the asymmetry of feature representations ΔD reduces the level of redundancy between the R_X and R_Y – so, as anticipated, increasing this asymmetry will increase the assignment error rate.

This redundancy represents the cost of our solution to the assignment problem. Thus, we will search for a solution to the assignment problem that achieves a particular assignment error rate and local distortion magnitude while using as few bits – and, in particular, as few redundant bits – as possible. Using the information constraint introduced here, we study how the assignment error rate depends on the number of commonly represented features and feature representation asymmetry for a fixed number of bits. This analysis reveals a trade-off between catastrophic assignment errors and local estimator variance that the brain must navigate when constructing distributed representations with limited resources.

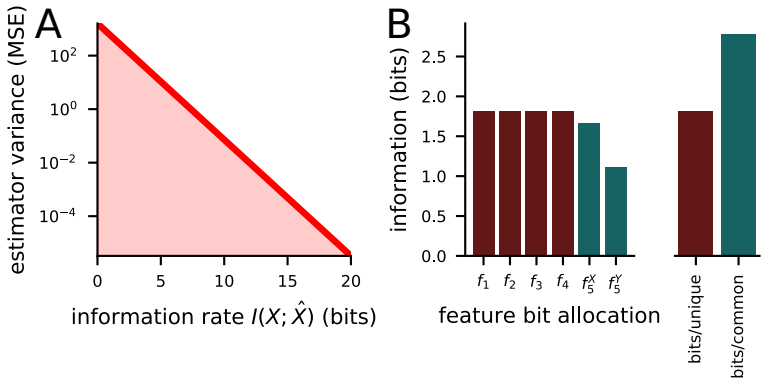

Figure 3.4: Rate-distortion theory provides a connection between represented information, redundancy, and estimator variance. **A** The approximate rate-distortion bound for a uniform source. The bound gives the minimum estimator variance possible for any representation of a uniformly distributed quantity that uses a particular number of bits. **B** Introducing a constraint on the total information used across R_X and R_Y means that bits have to be allocated to the representation of each unique feature (red) and each commonly represented feature in each region (blue-green). Here, the bits are allocated to achieve a uniform estimator variance D_S. To achieve that target estimator variance, the commonly represented features require many more bits than the uniquely represented features – this is due to their additional redundancy.

3.2.3 The trade-off between assignment errors and feature distortion

The representational system that we are studying makes two classes of errors, as we have described. The first kind of error is the non-local assignment error, in which some of the features of one object are integrated with features from a different object. These non-local errors are likely to be particularly catastrophic for behavior. The other kind of error is the local distortion of object representations, quantified by the mean squared-error of an optimal estimator. By introducing a constraint on the total amount of information used in the representational system, we discover a trade-off between these two kinds of errors. In particular, increasing the number of commonly represented features decreases the assignment error rate (Figure 4.5a, left), but increases the magnitude of local distortion (Figure 4.5a, right); while increasing the representation asymmetry increases the assignment error rate (Figure 4.5b, left), but decreases the magnitude of local distortion (Figure 4.5b, right).

We have already seen that increasing the number of commonly represented features C increases the magnitude of local errors (from Eq. 3.6 and see Figure 4.5a, left) and that increasing feature representation asymmetry decreases the magnitude of local errors (from Eq. 3.6 and see Figure 4.5a, right). However, the mechanics of assignment errors are more complicated. First, we recall that increasing the number of commonly represented features decreases the assignment error rate by making stimulus pairs at larger distances more probably (see Figure 4.2). However, we have just shown that increasing the number of commonly represented features under the information constraint causes both D_X and D_Y to increase, which is known to increase the assignment error rate. Thus, increasing the number of common features could either increase or decrease the assignment error rate under the information constraint. We find empirically that the decrease due to changes to the distribution of distances are orders of magnitude larger than the increase due to increased estimator variance D_X and D_Y (Figure 4.5b, left). Second, we make the impact of feature

representation asymmetry on assignment errors explicit, using our information constraint. For a fixed number of commonly represented features C, we recall that the lower bound on the assignment error rate increases with increases in the sum of estimator variances D_X and D_Y. The sum of the two estimator variances has the form given in Eq. 3.5, which we can now rewrite given Eq. 3.6 to depend explicitly on the other features of the representation system,

$$\begin{aligned} D_X + D_Y &= \frac{4}{1 - \Delta D^2} D_S \\ &= \left[\frac{4}{1 - \Delta D^2}\right]^{\frac{K}{K+C}} s^2 \exp\left(-\frac{2B}{K+C}\right) \end{aligned}$$

Here, we see that the dependence on representation asymmetry ΔD has flipped relative to Eq. 3.6. There, we showed that increasing representation asymmetry decreases D_S; here, we see that increasing representation asymmetry increases $D_X + D_Y$ (as before), even under the full information constraint. In particular, this means that, for a fixed C, increasing representation asymmetry ΔD will increase assignment errors under the information constraint (Figure 4.5b, right).

Next, we investigate this trade-off in the local distortion-assignment error rate plane (Figure 4.5c). For a fixed number of commonly represented features, changes in both the amount of information and the representation asymmetry describe a surface in the plane. This surface extends from zero to infinity along the information axis (down and to the left in Figure 4.5c) and from zero to one along the asymmetry axis (down and to the right in Figure 4.5c). The geometry of the surfaces in this space reveals a surprising result: For fixed information, the surfaces corresponding to many different numbers of commonly represented features (different colors, Figure 4.5c) have significant overlap. That is, many of the same assignment error and local distortion pairs can be achieved by several different numbers of commonly represented features and representation asymmetry for a fixed amount of information. Thus,

when minimizing weighted sums of assignment errors and local distortion, there will typically be several optimal solutions from a range of numbers of commonly represented features – and, as the number of commonly represented features increases, representation asymmetry will also increase. Thus, instead of considering an optimal pair of number of commonly represented features C and representation asymmetry ΔD, we consider an optimal level of redundancy. This works because redundancy directly implies both an assignment error rate and feature asymmetry (Figure 4.5d) with only minor deviations from the details of the representation. Thus, if a representation with a particular number of commonly represented features can achieve that level of redundancy, then it follows that there exists a level of representation asymmetry that achieves a particular assignment error and local distortion pair.

By constraining the total amount of information used in these distributed stimulus representations, we elucidate a global trade-off between redundancy for solving the assignment problem and efficiency for increasing the precision of those stimulus representations. Increasing the number of commonly represented features increases redundancy between the representations in X and Y: this reduces the assignment error rate, but decreases the precision of the individual stimulus feature representations; increasing feature representation asymmetry decreases redundancy between the representations in X and Y: this increases the precision of stimulus feature representations, but increases the assignment error rate. In constructing solutions to the assignment problem – within and across distinct sensory systems – the brain is likely to leverage both of these trade-offs. Further, solutions to the assignment problem across sensory modalities are likely to be constructed across evolutionary timescales and requirements for a particular level of redundancy in resultant information may be crucial to shaping sensory transducers.

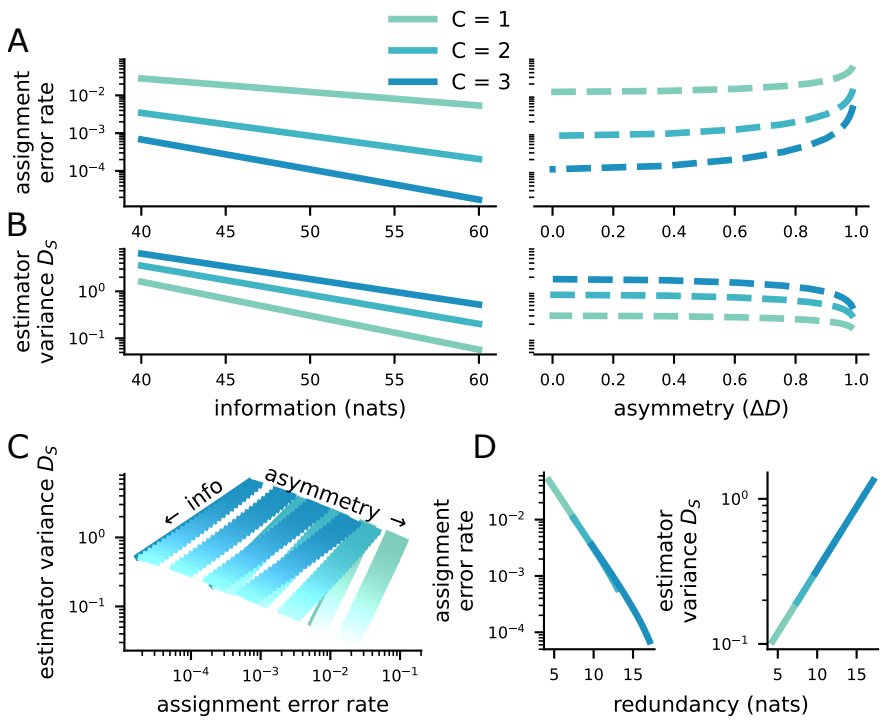

Figure 3.5: Constrained information gives rise to a tradeoff between assignment and local errors that is negotiated by both the number of commonly represented features and representation asymmetry. **A** For a fixed amount of information, increasing the number of commonly represented features C decreases the assignment error rate (left); increasing representation asymmetry increases the assignment error rate for all numbers of overlapping features (right). **B** For a fixed amount of information, increasing the number of commonly represented features C increases the magnitude of local errors (estimator variance D_S, (left); increasing representation asymmetry decreases the magnitude of local errors (estimator variance D_S, right). **C** The performance of different commonly represented features C and different representation asymmetries on the assignment error-local error plane. Here, the tradeoff is shown as increasing asymmetry moves codes down and to the right (decreasing local error magnitude, increasing assignment error rate) while increasing the number of commonly represented features moves codes up and to the left (increasing local error magnitude, decreasing assignment error rate). **D** The redundancy between the two representations specifies both an assignment error rate (left) and an estimator variance (right). The representation asymmetry (changes along the lines) and number of overlapping features (different colors) both affect the redundancy, but do not produce significant changes in either the assignment error rate or the estimator variance aside from that.

3.2.4 Predictions for behavior

Our framework also makes numerous predictions for how the frequency of assignment errors should scale in different contexts. We have tested several of these predictions in existing experimental data. This framework makes predictions for how assignment error, and – as a consequence – report error in general, should scale depending on the distance between multiple stimuli along two distinct classes of feature dimension: uniquely represented features and commonly represented features. Further, the rate-distortion bound makes a more generic prediction for how assignment errors should depend on the number of simultaneously represented stimuli.

To test these predictions, we used experimental data from a widely used class of tasks in the human psychophysical literature. In these tasks, an array of stimuli is presented to human participants, who are tasked with remembering the stimuli in the array. Then, after a delay period, a single stimulus from the array is cued, typically by showing its spatial position, and the human participant is tasked with reporting the other uncued features of that stimulus. As an example, in [160], an array of colored squares are presented as the sample, the participant is cued with a location corresponding to one of those squares, and, finally, the participant has to report the color of the square that was displayed at the cued location (Figure 4.6a). This is a different context than the one we have been discussing so far. However, our framework directly applies here as well. In particular, instead of the assignment problem being solved between two sets of representations expressed by anatomically distinct populations of neurons or in distinct sensory modalities, the assignment problem is being solved across two sets of representations that are separated in time. The first set of representations, in our example, has information about both spatial position and color of the squares; the second set of representations has information only about spatial position. Our theory governs the likelihood of assignment errors in both cases.

In particular, our theory makes several quantitative and qualitative predictions for errors on tasks like the one described above. First, our framework predicts that assignment error likelihood should depend on proximity in angular position (Figure 4.6b, short arrow) but not on proximity in color (Figure 4.6b, long arrow). Further, our framework also makes predictions about how errors should scale with set size. To test the predictions of our theory with these data, we fit three models. In the first, which we refer to as the assignment-only model, errors are explained only in terms of local distortion and assignment errors. In the second, the guessing-only model, errors are explained by local distortion and random, uniformly distributed guesses. These guesses are thought to occur due to a limited number of working memory slots. If the number of slots is lower than the number of stimuli and the target stimulus is not encoded in one of the slots, then the participant is assumed to randomly guess. In the third, the combined model, errors are explained in terms of local distortion, assignment errors, and guessing. The models are hierarchical across subjects and fit using Hamiltonian Monte Carlo (for more details see *Behavioral model fitting* in *Methods*). We evaluate goodness of fit using a Bayesian approximation of leave-one-out cross-validation referred to as PSIS-LOO[161, 162]. This analysis reveals that the combined model performs reliably better than either of the other models (combined against assignment-only: $\Delta\,\mathrm{ELPD}_{\mathrm{loo}} = 321 \pm 61$; combined against guessing-only: $\Delta\,\mathrm{ELPD}_{\mathrm{loo}} = 34 \pm 9$). While the combined model has one additional parameter per participant, PSIS-LOO accounts for increases in performance due to the inclusion of additional degrees of freedom. Further, the combined model predicts that the participants will make both assignment and guessing errors (Figure 4.6c).

Next, using the combined model, we evaluate how well the errors made by the human participants match the specific predictions made by our framework. First, we show that there is good agreement between the overall error distribution of the model and of each individual participant (Figure 4.6d). Next, we show that our framework matches how report error scales

with increased numbers of stimuli. In particular, report error increases monotonically with the number of stimuli, as has been widely observed[160], and our rate-distortion framework along with our model for the frequency of assignment and guessing errors matches the speed of that increase in the all participants (Figure 4.6e. Finally, we test a more diagnostic prediction: our framework also predicts that the assignment error rate should not strongly depend on proximity between stimuli in the reported feature – in this case, color. We show that our model provides a good fit to this relationship in these data as well (Figure 4.6f) – and that, indeed, the report error across different distances in the reported feature is relatively flat. Thus, in three important ways, our framework provides a quantitative fit to human performance on psychophysical memory tasks – even though the assignment problem here is being solved across time, rather than across feature space as we have previously discussed.

Our framework makes one additional key prediction that is mentioned above but that we have yet to quantitatively test. In particular, the likelihood of an assignment error should strongly depend on the distance between pairs of stimuli in the common feature (in the data described above, this common feature is angular position). That is, human participants should be more likely to make assignment errors when the cued stimulus and a distractor stimulus are nearby to each other. While we have not qualitatively tested this prediction yet, there is widespread support for it in the existing literature on feature integration theory[158] and psychophysical experiments on working memory[92]. However, our framework can potentially be used to understand the frequency of assignment errors in more complex feature spaces. For instance, assignment errors in feature integration theory have been shown to depend not just on spatial proximity, but also on proximity in other putatively common features[158]. Our framework can leverage this observation to predict differences, or asymmetries, in the representation of these different features across different involved brain regions. Thus, our framework provides a potentially useful tool for deepening our understanding of neural representations from only psychophysical observations.

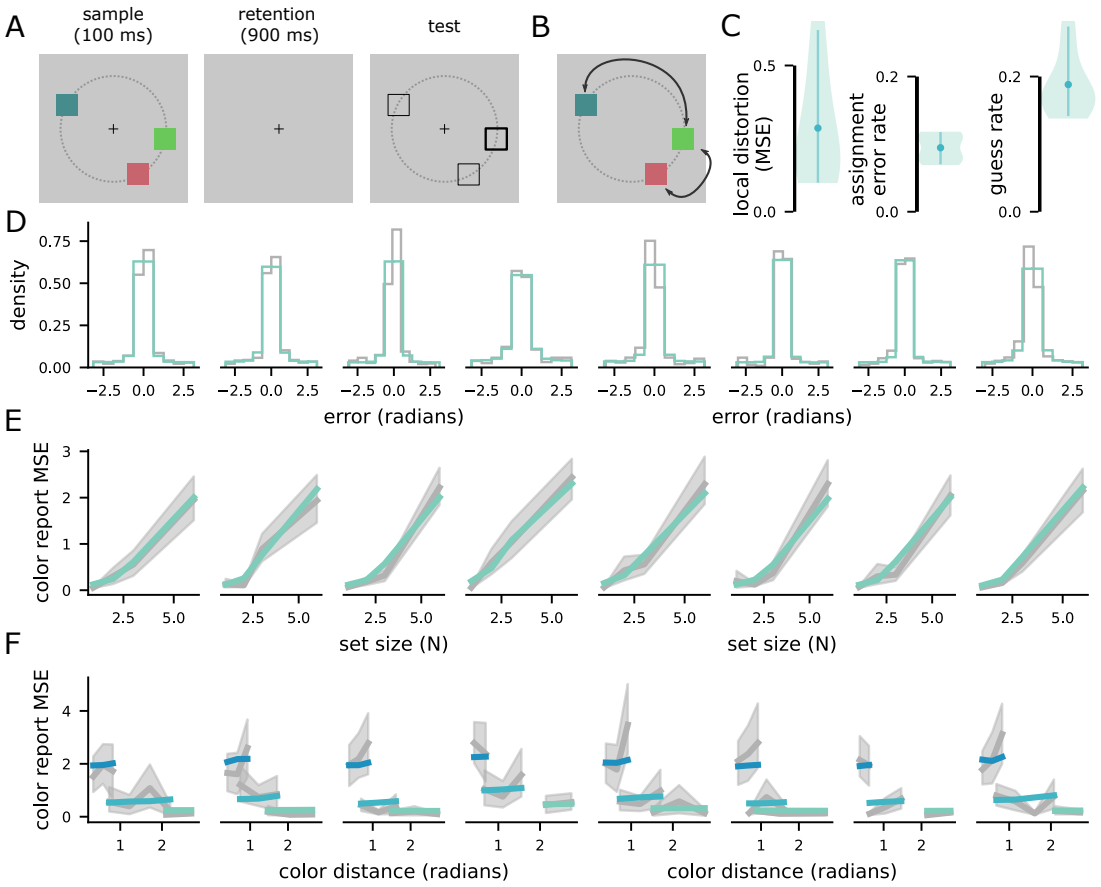

Figure 3.6: The framework for the assignment problem developed here fits experimental data in humans. **A** Schematic of the recall task. Human subjects were presented with 1, 3, or 6 colored squares at equal eccentricity from a central fixation point (dashed cirlce not shown to subjects). The squares were presented for 100 ms and the subjects were asked to remember all of them. Then, after a 900 ms delay period, a test array was presented with cues indicating the positions of all of the previously shown stimuli. The target stimulus was indicated by thick black lines, and the subjects were asked to report the color of that stimulus using a randomly oriented color wheel (not shown). **B** A schematic of likely and unlikely errors in our framework, using the example from **A**. Errors to stimuli that are nearby in position are more likely (short arrow) than errors to stimuli that are nearby in color (long arrow). **C** Violin plots showing the predicted local distortion (MSE, left), assignment error rate (center), and guess rate (right) of the participants in [160] when asked to remember four stimuli. There is significant diversity among participants. **D** A comparison of the error distribution produced by our model (color) and the error distribution of each participant (grey). **E** Our model predicts that report error will scale with set size. Model fit is in color; human subject data is in grey. **F** Our model predicts that report error will not strongly scale with the similarity of remembered colors. Model fit is in color; human subject data is in grey. The model fits well for most of the participants.

3.3 Discussion

We have described and analyzed a general solution to the representation assignment problem. Using this formalization, we illustrated a trade-off between local errors, quantified by estimator variance, and catastrophic assignment errors, which result in the illusory conjunction of subsets of stimulus features. We showed that assignment errors depend on the distance between stimuli in the shared representation space, and that increasing the dimensionality of that shared space will – on average – decrease their probability. Further, we showed many weakly overlapping stimulus features can provide a reliable solution to the assignment problem that finds a compromise between efficiency, for reducing estimator variance, and redundancy for reducing the probability of assignment errors. Finally, we showed that the pattern of local and assignment errors in experimental data from human working memory experiments matches several of the predictions of our framework, even though, in this case, the assignment is performed between two sets of representations that are spread across time rather than across different populations of neurons, which is the context we have focused on in the majority of this work. This indicates the generality of our solution to and theory of the assignment problem.

One further test of our framework will be to quantify the level of redundancy in information represented across distinct brain regions. Recent large scale recordings of multiple brain regions with rich stimulus sets provide a fertile ground for quantifying the amount of redundant and unique information represented in many brain regions, while previous datasets were limited due to either sparse samples of the neural population or sparse samples of potential stimulus sets. This framework makes predictions about the balance between redundant and unique stimulus information that produces both a reliable solution to the assignment problem and efficient stimulus representations. This new experimental data in conjunction with the framework we develop here can serve as building blocks for a comprehensive theory

of the mesoscale organization of object representations in the brain. In particular, previous theoretical work on neural representations has focused on population codes within single brain regions[51, 62, 102–104]. While understanding these codes is extremely important, it is also important to understand neural codes contextualized within the distributed, modular representations that have been experimentally observed in the brain for decades. While modular representations have some clear intuitive benefits (such as), they have received relatively little focused theoretical treatment. This framework illuminates a tension inherent to modularity: That ambiguity arises in the presence of multiple stimuli, and this ambiguity can only be resolved through redundant representations, counter to within-region priorities suggested by the theory of efficient coding[46, 47].

Further, neural codes for multiple stimuli have also received relatively little attention. Ambiguities in assignment primarily arise in the presence of multiple stimuli, which is not well characterized in either the experimental or theoretical literature. Here, we have described the solution to one difficulty in distributed multi-object representations. However, further work will need to incorporate a solution to the clustering problem that we elide here: the number of distinct representations must also be inferred from neural representations, which is another facet of the binding problem. Further, how these representations interact with each other within each region also requires further study. Previous theoretical work has outlined how the encoding of multiple stimuli in a single population of neurons can interfere with each other, and how different mixing strategies can either alleviate or exacerbate that interference[163]. However, correlation structure in neural responses within a region can also be an effective mechanism for reducing the probability of assignment errors. In particular, positively coordinated feature estimates across stimuli within a single region can guarantee that no assignment errors will occur even for large estimator variance. However, it remains unclear how combinations of neural noise correlations, stimulus tuning, and representation mixing strategies can produce this positive correlation in estimates. Deriving these condi-

tions may provide a novel explanation for noise correlations and mixing strategies in the absence of attention.

Finally, we have focused our analysis here on an explicit redundancy between distributed representations, as arises from the representation of the same or similar stimulus features (even if they are in distinct frames of reference, as with position estimates in the auditory and visual systems). However, future work will probe the role of implicit, learned redundancy between particular feature values. Intuitively, this redundancy will be essential for making associations between olfactory percepts, which have relatively little position or otherwise explicit redundant information with other senses, and other sensory percepts. While implicit redundancy must be learned, it could be learned quickly when coupled with explicit redundancy – fast learning of implicit associations by leveraging relatively small levels of explicit redundancy may be an alternate mechanism by which the brain can rapidly and generally solve the assignment problem with potentially more efficient overall representations. Further work can probe this trade-off.

In summary, we have developed a general framework for understanding how the brain makes sense of distributed representations of multiple objects – that is, how the brain solves the representation assignment problem, by overcoming an ambiguity inherent in those representations if given perfectly efficient and non-redundant representations. This framework not only provides an explanation for human behavior in related experimental tasks, but also points to many new directions in the study of neural codes: directions that explicitly contextualize neural representations in a distributed network of brain regions as well as move toward understanding how neural codes can be made suitable for encoding heterogeneous, multi-object environments.

Acknowledgments: This work was supported by NIH F31EY029155 (WJJ),
NIH R01EY019041 (DJF), CRCNS NIH R01MH115555 (DJF), NSF NCS 1631571 (DJF),

and a DOD Vannevar Bush Fellowship (DJF).

Author contributions: WJJ conceived of the project. DJF supervised the project development. WJJ created the model and performed the calculations, model simulations, and data analysis. WJJ and DJF wrote the paper.

Competing interests: The authors declare no competing interests.

3.4 Methods

3.4.1 Definition of the objects

An object set S of size N is N independent samples from the multidimensional probability distribution $p(s)$ – thus, $p(S) = \prod_i^N p(s_i)$. Each s_i is a vector of length K, so $p(s)$ is a probability distribution over a K-dimensional space. Each of these dimensions represents a feature of the object, such as color, spatial frequency, pitch, or orientation. In the majority of the paper, we assume that the objects are uniformly distributed in the space – that is, each point in the K-dimensional volume is equally likely to occur.

Changes in the distribution of the objects does not produce a qualitative change in our results, though it does affect the balance between assignment errors and the magnitude of local distortion (see).

3.4.2 Definition of the representations

We focus on the representation of our K-dimensional objects in two brain regions, R_X and R_Y. We assume that both regions have some common and some unique information. That is, neither region encodes all K object features. This is guaranteed to be the case when R_X and R_Y represent early sensory areas from different sensory modalities – and there is evidence that a balance of unique and common information is preserved across hierarchies

of sensory brain regions even within single modalities, such as in the primate visual system.

Formally, R_X encodes a subset of the K total object features, denoted as F_X – and similarly for R_Y. Each of the features can be identified by their index from 1 to K, and region R_X is said to encode feature $i \in \{1, 2, ..., K\}$ with local distortion D_X^i, which is the variance of an optimal estimator for the value of feature i from the neural activity in region R_X – and, again, similarly for R_Y.

Thus, the subset of features represented in R_X that are not represented in R_Y are the unique information from R_X – that is, $F_X \setminus F_Y$. When R_X is an auditory region and R_Y is a visual region, then these unique features might include representations of pitch and timbre. Further, the subset of features represented in both R_X and R_Y are the common information, which is essential for solutions to the assignment problem. The size of this intersection $|F_X \cap F_Y| = C_{XY} = C$ has important consequences for the assignment error rate, and the achievable local distortion when the representation capacity is constrained.

Here, we study the reliability of inferences about the original set of N objects S, each of which are described by K feature dimensions, from the neural activity in two distinct regions, R_X and R_Y, which both encode some common and some unique information about the objects. Inferring the original object set from two distinct representations requires the two sets of representations to be combined with each other, which we refer to as assignment. After assignment, information about the commonly represented features can be combined according to its precision. That is, if feature i is represented with local distortion D_X^i in region R_X and D_Y^i in region Y, then the combined estimate of feature i from both regions (assuming correct assignment) will be,

$$D_S^i = \frac{D_X^i D_Y^i}{D_X^i + D_Y^i}$$

where D_S^i, in general, refers to the local distortion for feature i in the object set inferred from representations in both R_X and R_Y.

3.4.3 Definition of assignment errors

As mentioned above, to make inferences about the whole object set from two distinct sources of information (i.e., R_X and R_Y), the two sets of representations must be integrated with each other. When there is only one object ($N = 1$), then this integration is trivial as there is only one possible one-to-one mapping between the two sets of representations, and this mapping is correct. However, when there is more than one object ($N > 1$), then assignment errors become possible. In particular, for N objects, there are $N!$ possible one-to-one mappings (i.e., assignments) between the two sets of representations, $\hat{X}$ and $\hat{Y}$. Thus, if there is no information about which mapping to select, assignment errors have the probability $1 - 1/N!$, which is near 1 for even relatively small N.

Formally, we can frame the mapping selection problem as an inference about which of the possible maps M is most likely to account for the observed representations $\hat{X}$ and $\hat{Y}$ in R_X and R_Y, respectively. This can be written as,

$$\begin{aligned} p(M|\hat{X},\hat{Y}) &= \frac{p(\hat{X},\hat{Y}|M)p(M)}{p(\hat{X},\hat{Y})} \\ &\propto p(\hat{X}|\hat{Y},M)p(\hat{Y}|M) \\ &\propto p(\hat{X}|\hat{Y},M) \end{aligned} \tag{3.7}$$

where we proceed by first assuming that the prior probability of each map is the same (i.e., $p(M)$ is uniform) and that the representations in one region do not depend on the map (i.e., $p(\hat{Y}|M) = p(\hat{Y})$, and similarly for $\hat{X}$). Thus, we are left with a single term, $p(\hat{X}|\hat{Y},M)$, that gives the likelihood of the representations observed in one region (here, $\hat{X}$, but X and Y are

interchangeable) conditioned on a particular map M and the representations observed in the other region ($\hat{Y}$). If $\hat{X}$ and $\hat{Y}$ are independent of each other, then $p(\hat{X}|\hat{Y}, M) = p(\hat{X})$, as above, and all maps are equally likely. So, it is dependence between the representations in R_X and R_Y that enables the correct assignment to be selected at a rate better than chance. Thus, dependence between those representations is necessary for a reliable solution to the assignment problem. In general, this observation already indicates to us that we should expect pairs of brain regions to encode some common information – so that the assignment problem can be solved – and some unique information – due to both distinct sensory systems, but also due to considerations related to efficient coding, which we make explicit below.

Probability of assignment errors

Given the above inference process, we can now characterize the likelihood that an assignment error occurs given different levels of common information shared between $\hat{X}$ and $\hat{Y}$. From our formalization of the representations above, we know that each object feature i is estimated from the activity of neurons in region R_X with variance D^i_X, and similarly for R_Y. Here, we assume that these estimates are unbiased and Gaussian distributed, with mean equal to the true value of the object feature and variance as given. The Gaussian distribution is the maximum entropy distribution for fixed mean and variance, which means that these estimates will contain less information about the true value of the object feature than any other distribution with the same mean and variance – thus, this assumption represents an upper bound on the difficulty of the integration task.

Using this formalization, we can write an explicit form for Eq. 3.7,

$$p(M|\hat{X},\hat{Y}) \propto \prod_{i}^{N}\prod_{j}^{C} \exp\left(-\frac{(\hat{x}_i^j - M_i\hat{y}^j)^2}{2(D_X + D_Y)}\right)$$
$$\log(p(M|\hat{X},\hat{Y})) \propto -\sum_{i}^{N}\sum_{j}^{C}(\hat{x}_i^j - M_i\hat{y}^j)^2 \tag{3.8}$$

where $\hat{x}_i^j$ and $\hat{y}_i^j$ are the values of feature j for object i from the $N \times C$ estimation matrix of the C commonly represented features for each of the N objects. The common features are ordered consistently across $\hat{x}$ and $\hat{y}$, while the N different objects are not. The map M is an $N \times N$ permutation matrix. Thus, in finding the most likely map, we would search over the $\binom{N}{2}$ possible permutation matrices to find the one that maximizes Eq. 3.8. This is equivalent to finding the permutation matrix that minimizes the sum squared distance between integrated object representation pairs in the C-dimensional shared feature space.

From the above, it follows that an assignment error occurs precisely when the representation of two objects cross over each other in one of the two brain regions, but not in both – as schematized in Figure 4.1c, red arrow and assignment lines. For the commonly represented feature values of two objects, x_1^c and x_2^c, the probability that this crossover happens in R_X depends on the distribution of the distances between their estimates,

$$\hat{x}_2^c - \hat{x}_1^c \sim \mathcal{N}(\delta, 2D_X)$$

where δ is the distance between the true values of x_1^c and x_2^c (i.e., $\delta = x_2^c - x_1^c$) and we assume, without loss of generality, that $x_1^c < x_2^c$. Since δ is still Gaussian distributed, we write the probability that the estimate of the first object becomes greater than the estimate

of the second object (i.e., that $\hat{x}_1^c > \hat{x}_2^c$) as

$$P(\text{cross in } R_X) = Q\left(\frac{-\delta}{\sqrt{2D_X}}\right)$$

where $Q(.)$ is the cumulative distribution function for the standard Gaussian distribution.

Following this, the full probability of an assignment error incorporates the probability that the cross occurs in R_X or R_Y as well as that it occurs in both (which would not result in an assignment error). We write this probability as,

$$F(\delta) = Q\left(\frac{-\delta}{\sqrt{2D_X}}\right) + Q\left(\frac{-\delta}{\sqrt{2D_Y}}\right) - Q\left(\frac{-\delta}{\sqrt{2D_X}}\right) Q\left(\frac{-\delta}{\sqrt{2D_Y}}\right)$$

where the final term is the probability that a cross occurs in both regions. While we have discussed a single common feature here, this expression is general, and applies for any value of C, so long as the local distortion D_X and D_Y for all of the common features is the same within each region, which we assume in the majority of the text. This expression already gives us insight into how assignment errors depend on D_X and D_Y for stimuli at some distance δ in a common feature space. However, in general, assignment errors also depend on how likely it is that two stimuli at a particular distance will be observed – that is, on $p_C(\delta)$. In the main text, we develop this dependence, as well as a dependence on the number of objects, which results in Eq. 3.1.

Assignment error rate approximation for $C = 1$

For one overlapping object feature ($C = 1$), we derive an approximate closed form for Eq. 3.1. The derivation is as follows,

$$\begin{aligned}
AE_1 &\leq \binom{N}{2} \int_0^s d\delta \, p_C(\delta) \, F(\delta) \\
&= \binom{N}{2} \int_0^s d\delta \, \frac{2(s-\delta)}{s^2} \left(Q\left(\frac{-\delta}{\sqrt{2D_X}}\right) + Q\left(\frac{-\delta}{\sqrt{2D_Y}}\right) - Q\left(\frac{-\delta}{\sqrt{2D_X}}\right) Q\left(\frac{-\delta}{\sqrt{2D_Y}}\right) \right) \\
&\approx \binom{N}{2} \int_0^s d\delta \, \frac{2(s-\delta)}{s^2} \left(Q\left(\frac{-\delta}{\sqrt{2D_X}}\right) + Q\left(\frac{-\delta}{\sqrt{2D_Y}}\right) \right) \\
&= \binom{N}{2} \frac{2}{s^2} \int_0^s d\delta \, (s-\delta) Q\left(\frac{-\delta}{\sqrt{2D_X}}\right) + (s-\delta) Q\left(\frac{-\delta}{\sqrt{2D_Y}}\right)
\end{aligned}$$

since the two terms in this sum are analogous to each other, we deal with them separately before combining.

So, for the first part of each term,

$$\begin{aligned}
\int_0^s d\delta \, s \, Q\left(\frac{-\delta}{\sqrt{2D_i}}\right) &= -s\sqrt{2D_i} \left(-\frac{\delta}{\sqrt{2D_i}} Q\left(-\frac{\delta}{\sqrt{2D_X}}\right) + \phi\left(-\frac{\delta}{\sqrt{2D_X}}\right) \right) \Big|_0^s \\
&= -s\sqrt{2D_i} \left(\phi\left(-\frac{s}{\sqrt{2D_i}}\right) - \frac{s}{\sqrt{2D_i}} Q\left(-\frac{s}{\sqrt{2D_i}}\right) \right) + s\sqrt{\frac{D_i}{\pi}} \\
&\approx s\sqrt{\frac{D_i}{\pi}}
\end{aligned}$$

where ϕ is the standard normal density function and the approximation in the last line holds when $s \gg D_i$, which is the regime we focus on for the main text.

Now, for the second part of each term,

$$
\begin{aligned}
-\int_0^s d\delta\, \delta\, Q\left(\frac{-\delta}{\sqrt{2D_X}}\right) &= -4D_i \left(\left(\frac{\delta^2}{2D_i} - 1\right) Q\left(-\frac{\delta}{\sqrt{2D_i}}\right) - \frac{\delta}{\sqrt{2D_i}}\, \phi\left(-\frac{\delta}{\sqrt{2D_i}}\right)\right)\Big|_0^s \\
&= -4D_i \left(\left(\frac{s^2}{2D_i} - 1\right) Q\left(-\frac{s}{\sqrt{2D_i}}\right) - \frac{s}{\sqrt{2D_i}}\, \phi\left(-\frac{s}{\sqrt{2D_i}}\right)\right) - 2D_i \\
&\approx -2D_i
\end{aligned}
$$

where, again, the approximation in the last line holds when $s \gg D_i$.

Then, combining the two expressions above,

$$
\frac{2}{s^2}\int_0^s d\delta\, (s-\delta)\, Q\left(-\frac{\delta}{\sqrt{2D_i}}\right) \approx \frac{2}{s}\sqrt{\frac{D_i}{\pi}} - \frac{4}{s^2} D_i
$$

Before, finally, combining the terms corresponding to D_X and D_Y gives,

$$
\begin{aligned}
AE_1 &\approx \binom{N}{2}\left[\frac{2\sqrt{D_X}}{s\sqrt{\pi}} - \frac{4D_X}{s^2} + \frac{2\sqrt{D_Y}}{s\sqrt{\pi}} - \frac{4D_Y}{s^2}\right] \\
&= \binom{N}{2}\left[\frac{2\sqrt{D_X + D_Y}}{s\sqrt{\pi}} - \frac{4(D_X + D_Y)}{s^2}\right] \\
&\approx \binom{N}{2}\frac{2\sqrt{D_X + D_Y}}{s\sqrt{\pi}}
\end{aligned}
$$

This final form is given as Eq. 3.2 in the main text.

3.4.4 Limits on representation capacity

Our limit for representation capacity is provided by the rate-distortion bound, which gives a lower bound on the achievable magnitude of local distortion (MSE) given the distribution of the input and a particular number of bits of information used in the representation. That is, the rate-distortion bound provides the smallest achievable mean squared error when B bits are used in the representation. First, we derive an approximate rate-distortion bound

for the uniform object distribution that we consider here, as well as one for the Gaussian object distribution that we also consider (and show that they are proportional to each other in our conditions of interest). Then, we use the bound the show a dependence between the number of commonly represented features C, the number of objects N, and the magnitude of local distortion in each region D_X and D_Y. Finally, we show that this dependence leads to a trade-off between local distortion and assignment errors.

Rate-distortion bound for a uniform object distribution

To derive the rate-distortion bound in this context, we consider a source X, which is uniformly distributed on an interval $[0, s]$ and an estimate of that source $\hat{X}$ derived from a noisy version of X. The distortion metric that we consider here is the mean squared error (MSE), as discussed in the rest of the paper. So, the rate-distortion function $R(D)$ is,

$$R(D) = \min_{f(\hat{x}|x):E(\hat{X}-X)^2 \leq D} I(X;\hat{X})$$

where $I(X;\hat{X})$ is the mutual information between X and $\hat{X}$ and $f(\hat{x}|x)$ is a function mapping from the original input x to its estimate $\hat{x}$. So, we want to find the $f(.|.)$ that minimizes the amount of information $I(X;\hat{X})$ while achieving a fixed mean squared error D. To proceed, we first notice that having fixed boundaries in our input distribution will likely distort the output distribution when s and $\sqrt{D}$ are similar in magnitude. Thus, we develop an approximate rate-distortion bound for when $s \gg \sqrt{D}$ and that boundary condition does not affect our results. We also consider Gaussian distributed inputs, which do not have the same boundary condition, and show that our results do not qualitatively change.

Proceeding when $s \gg \sqrt{D}$,

$$
\begin{aligned}
I(X;\hat{X}) &= H(X) - H(X|\hat{X}) \\
&= \log(s) - H(X - \hat{X}|\hat{X}) \\
&\text{subtracting a constant does not change the entropy} \\
&\leq \log(s) - H(X - \hat{X}) \\
&\text{since conditioning always reduces entropy} \\
&\leq \log(s) - H(\mathcal{N}(0, E(X - \hat{X})^2)) \qquad (3.9) \\
&\text{since the maximum entropy distribution for fixed variance on} \\
&\text{unconstrained support is Gaussian} \\
&= \log(s) - \frac{1}{2}\log(2\,\pi\,D) \\
&= \frac{1}{2}\log\frac{s^2}{2\,\pi\,D}
\end{aligned}
$$

where Eq. 3.9 is only a valid approximation when $s \gg \sqrt{D}$ as commented on above. Now, we must show that this lower bound is achievable (again, with the condition above). For this, we consider the channel,

$$\hat{X} = X + Z \qquad (3.10)$$

where $Z \sim \mathcal{N}(0, D)$. Thus, we can evaluate the mutual information,

$$\begin{aligned} I(X;\hat{X}) &= H(\hat{X}) - H(\hat{X}|X) \\ &\approx \log(s) - H(\hat{X}|X) \\ &\text{due to } s \gg \sqrt{D} \\ &= \log(s) - \frac{1}{2}\log(2\,\pi\,D) \\ &\text{by Eq. 3.10 and the definition of } Z \\ &= \frac{1}{2}\log\frac{s^2}{2\,\pi\,D} \end{aligned}$$

which shows that this simple channel can achieve our bound.

The full rate-distortion bound also must account for the case when $s^2 < 2\,\pi\,e\,D$, for which we assume a deterministic mapping from X to $\hat{X}$ and, thus, set $R(D) = 0$. So,

$$R(D) \approx \begin{cases} \frac{1}{2}\log\frac{s^2}{2\,\pi\,D} & s^2 \geq 2\,\pi\,D \geq 0 \\ 0 & s^2 < 2\,\pi\,D \end{cases}$$

Finally, we will use the inverse of this function throughout the paper, which is referred to as the distortion-rate bound $D(R)$. We derive it here,

$$\begin{aligned} B = R(D) &= \frac{1}{2}\log\frac{s^2}{2\,\pi\,D} \\ \exp(2B) &= \frac{s^2}{2\,\pi\,D} \\ D &= \frac{s^2}{2\,\pi}\exp(-2B) \end{aligned} \tag{3.11}$$

which is, again, subject to the same range constraints given above and valid only under the same condition.

Rate-distortion bound for a Gaussian object distribution

The derivation of the rate-distortion bound for a Gaussian object distribution (i.e., Gaussian-distributed feature values) is very similar to the above, and is given in detail in [61]. So, we do not go into detail here.

The rate-distortion bound for a Gaussian input $X \sim \mathcal{N}(0, s^2)$ is given by,

$$R(D) = \begin{cases} \frac{1}{2} \log \frac{s^2}{D} & 0 \leq D \leq s^2 \\ 0 & D > s^2 \end{cases}$$

and can be inverted to obtain,

$$D = s^2 \exp(-2B)$$

where B is the amount of bits used in the representation, as above. We note that this expression is proportional to Eq. 3.11.

Representation capacity and the rate-distortion bound

Now, we can develop a notion of representation capacity for each brain region or, generally, each pair of brain regions. In particular, we can constrain both R_X and R_Y to use precisely B bits of information to represent the objects S. Then, using the rate-distortion bound developed above as well as the other parameters of our representations (the number of commonly represented features and representation asymmetry), we can derive an expression for the local distortion (MSE) achieved by the integrated representations.

To proceed, we abstract both the uniform and Gaussian input rate-distortion bounds from

above, writing instead,

$$D = \frac{s^2}{p} \exp(-2B)$$

and

$$B = \frac{1}{2} \log \frac{s^2}{pD}$$

where $p = 2\pi$ if the input is uniform-distributed and $p = 1$ if it is Gaussian-distributed. Next, we assume that the total number of bits will be allocated evenly among the encoded objects – as indicated by some psychophysical experiments[160, 164] (but see [165]) – and that each feature is encoded so that the resulting feature estimate (drawing from both regions if it is commonly encoded) is constant across all features. That is, all of the features, once integrated successfully are encoded with local distortion (MSE) D_S.

With this structure, we derive an expression for D_S that depends on the number of objects N, the number of features K, the number of commonly represented features C, the feature representation asymmetry ΔD, and the bits of information B that are used to encode all of

this information. First, we derive this dependence terms of the total number of bits B,

$$\begin{aligned}
B &= N(K-C)\frac{1}{2}\log\frac{s^2}{pD_S} + NC\frac{1}{2}\log\frac{s^2}{pD_X} + NC\frac{1}{2}\log\frac{s^2}{pD_Y} \\
&= N(K-C)\frac{1}{2}\log\frac{s^2}{pD_S} + NC\frac{1}{2}\log\frac{s^4}{p^2D_XD_Y} \\
&= N(K-C)\frac{1}{2}\log\frac{s^2}{pD_S} + NC\frac{1}{2}\log\frac{s^4(1-\Delta D)(1+\Delta D)}{p^24D_S^2} \\
&\text{by eqs. (3.3) and (3.4)} \\
&= N(K-C)\frac{1}{2}\log\frac{s^2}{pD_S} + NC\frac{1}{2}\log\frac{s^4(1-\Delta D^2)}{p^24D_S^2} \\
&= NK\frac{1}{2}\log\frac{s^2}{pD_S} + NC\frac{1}{2}\left(\log\frac{s^4(1-\Delta D^2)}{p^24D_S^2} - \log\frac{s^2}{pD_S}\right) \\
&= NK\frac{1}{2}\log\frac{s^2}{pD_S} + NC\frac{1}{2}\log\frac{s^2(1-\Delta D^2)}{p4D_S} \\
&= (NC+NK)\frac{1}{2}\log\frac{s^2}{pD_S} + NC\frac{1}{2}\log\frac{1-\Delta D^2}{4}
\end{aligned}$$

Next, we solve this expression for D_S,

$$\begin{aligned}
B &= (NC+NK)\frac{1}{2}\log\frac{s^2}{pD_S} + NC\frac{1}{2}\log\frac{1-\Delta D^2}{4} \\
\exp\left(\frac{2B}{NK+NC}\right) &= \frac{s^2}{pD_S}\exp\left(\frac{NC}{NC+NK}\log\frac{1-\Delta D^2}{4}\right) \\
\exp\left(\frac{2B - NC\log\frac{1-\Delta D^2}{4}}{NK+NC}\right) &= \frac{s^2}{pD_S} \\
D_S &= \frac{s^2}{p}\exp\left(\frac{-2B + NC\log\frac{1-\Delta D^2}{4}}{NK+NC}\right) \\
&= \frac{s^2}{p}\exp\left(-\frac{2B}{NK+NC}\right)\left[\frac{1-\Delta D^2}{4}\right]^{\frac{C}{K+C}}
\end{aligned}$$

which is valid so long as $B > \frac{KN}{2}\log\frac{1-\Delta D^2}{4}$, due to the dependence of the rate-distortion bound above. This equation is also reproduced in the main text, Eq. 3.6. From this ex-

pression, we can already see several features of our formalization. First, we can see that increased feature asymmetry will strictly decrease D_S, as the single term that depends on D_S is decreasing in ΔD. Second, we can see that increasing C will strictly increase D_S whenever the equation is valid. To show this, we take the derivative of D_S with respect to C and show that it is positive after the condition above is met.

3.4.5 Behavioral model fitting

To adapt our framework to the experimental data discussed in the text, we made several assumptions about the sequence of events during the experiment. First, during the task delay period, we assume that each of the remembered representations undergoes a random, Gaussian drift, similar to that reported for other working memory representations[106]. Then, on the presentation of the test cue, we assume that any noise in the representation of the cue location is negligible compared to the noise in the remembered positions due to drift during the delay period. Formally, this is equivalent to assuming that,

$$F(\delta) \approx Q\left(\frac{-\delta}{\sqrt{2D_{\text{mem}}}}\right) + Q\left(\frac{-\delta}{\sqrt{2D_{\text{cue}}}}\right)$$
$$F(\delta) \approx Q\left(\frac{-\delta}{\sqrt{2D_{\text{mem}}}}\right)$$

where D_{mem} is the variance in position estimates of the remembered stimuli and D_{cue} is the variance of the estimate of cue position.

Further, because the data did not include the spatial positions of the presented stimuli, we included uncertainty about this in our model. That is, we knew that stimuli were presented in a subset of eight equally spaced positions at equal eccentricity[160]. Thus, we accounted for the probability that positions close to the recalled stimulus were occupied at each of the different set sizes.

In all our models, we fit two of the same parameters: the number of bits used to encode color and the error in the response introduced by the mechanics of the response (i.e., inherent discretization due to use of a mouse and keyboard). In the assignment-only model, we fit one additional parameter: the number of bits used to encode position, which, along with our model of stimulus position, determines the assignment error rate. In the guessing-only model, we fit a remembered stimuli parameter. That is, we assumed that the participants had a finite number of slots to remember a random subset of the stimuli in. The number of stimuli remembered on a given trial was assumed to be Poisson-distributed around the remembered stimuli rate parameter. If the participant did not remember the target stimulus, then we assumed they would randomly guess. Finally, for the combined model, we incorporated both of these elements. Assignment errors could only occur between the subset of remembered stimuli.

To fit our models to the experimental data from [160], we used the Bayesian probabilistic programming language Stan[166]. Thus, we formalized our three models in this programming language and then drew samples from the posterior distributions over our parameters given the observed experimental data. To account for both the variability between subjects and their broad similarity to each other, we used a multilevel model[167, 168]. In particular, this means that, while we fit each model parameter to each individual subject, we assumed that those parameters were themselves drawn from some distribution over subjects. We fit the parameters of those distributions as well. This hierarchical approach is an established way of partial pooling, where data from different sources are too heterogeneous to directly combine, but similar enough that better inferences can be gained from considering all of the data at once.

CHAPTER 4

THE LATERAL INTRAPARIETAL AREA IS PREFERENTIALLY ENGAGED IN DIRECTED TASKS RATHER THAN UNDIRECTED FREE BEHAVIOR

W. Jeffrey Johnston, Stephanie M. Tetrick, and David J. Freedman

Abstract

Reactions to the sensory world depend on context. For instance, explicit directions to search for a particular object or feature will induce different treatment of the sensory world than active exploration without a fixed goal. Understanding both directed search and undirected exploration is crucial to our understanding of the brain. However, our knowledge about these processes is typically derived from directed tasks, in which animals are given explicit cues as to which stimulus they should select. Here, we compare neural activity in the lateral intraparietal area (LIP) – which is thought to play a prominent role in both directed and undirected search behaviors – across two different tasks. In one task, the animal is given an explicit direction to select one of two stimuli from an array; in the other, the animal is allowed to choose a stimulus freely, though we manipulate the relevance of the stimuli to the animal. This comparison reveals that LIP is preferentially involved in the directed rather than undirected task. In particular, encoding of both the behavioral response and image relevance emerge earlier and are stronger in the directed relative to the undirected task. More generally, these results indicate the necessity of studying more naturalistic tasks when making inferences about the neural substrates of less constrained behaviors.

Keywords: lateral intraparietal area, context-dependence, electrophysiology, bottom-up and top-down attention

4.1 Introduction

The same sensory input can have vastly different semantic meaning and urgency depending on the context in which it is received. For instance, a hiker might not ordinarily notice or react to small trickles of water seeping out of nearby rock; however, if the hiker is out of water, those same seeps of water become both extremely salient and evoke a strong behavioral response – as they might indicate a spring of fresh, potable water. This kind of search typifies directed behavior. The hiker has a particular need, and they are searching their environment for ways to satisfy that need. Explicit search like this relies on the top-down deployment of attention[169, 170]. On the other hand, some items in the visual scene have features that are inherently salient. As an example, the sudden fall of a branch in the distance will likely capture the hiker's attention, because of its rapid onset and sharp contrast with the relative stillness of the forest. In this case, the visual features of the branch become salient due to their contrast with the surroundings[169, 170]. However, a novel type of bird perched in the bushes will likely also capture the hiker's attention. In this case, the bird need not draw a strong visual contrast with its surroundings to draw attention. Instead, its salience comes from its novelty – as many animals, including humans, prefer to orient toward novel rather than familiar stimuli[171–174]. In the absence of explicit goals, behavior entrained by these two forms of inherent salience – contrast and novelty – is referred to as undirected.

High contrast with nearby stimuli has been shown to produce a "pop-out" effect in visual search, where search time does not depend on the number of distractors. This "pop-out" is taken to indicate a bottom-up process for allocating visual attention[169, 170]. Further, psychophysical work in humans indicates that searches among highly familiar stimuli can exhibit "pop-out" effects as well[175], but that this depends primarily on the familiarity of the distractors rather than a difference in familiarity between the distractors and the target[176, 177]. Whether familiarity can induce "pop-out" effects in other contexts remains unclear. In

this work, we classify attentional capture by familiarity as a form of bottom-up attention (but not "pop-out" per se), due to the extensively documented and completely untrained ability of novel stimuli to capture attention relative to familiar stimuli[171–174]. In general, our understanding of the neuronal mechanisms of attention and the definitions of concepts like bottom-up and top-down attention have primarily relied on findings from directed behavioral paradigms, such as explicit search tasks[99, 178–180] and change detection tasks[181, 182]. Here, we ask whether the insights we have gained from these highly trained and directed behavioral paradigms generalize to an untrained and undirected context.

Previous work on bottom-up and top-down attention have demonstrated the involvement of a constellation of brain regions, spanning both the parietal and frontal cortices. In particular, the lateral intraparietal area (LIP)[26, 27, 99, 183] in the posterior parietal cortex (PPC) and the frontal eye fields (FEF)[99, 178, 183, 184] as well as regions in the dorsolateral prefrontal cortex (dlPFC), are thought to play important roles in the allocation of both overt (e.g., through an eye movement)[99, 185] and covert[185–188] spatial attention. In particular, these regions are thought to implement topographical maps of the sensory environment, where increased activity at a particular location in the map implies increased attention to that spatial location in the sensory world[27, 183]. Further, a study designed to discover the distinct roles of these regions in the allocation of attention showed that LIP is more directly involved in the bottom-up allocation of spatial attention, particularly in the context of a visual search task with "pop-out" visual features, while FEF is more directly involved in top-down deployment of attention, particularly in serial visual search tasks[99]. However, both of these tasks were highly trained and directed[99]; that is, to receive reward in both cases, the animal had to make an eye movement to a particular stimulus.

Other work in LIP has shown that its representations are strongly shaped by training[189–191]. In particular, over the course of training on a complex task, LIP has been shown to

develop strong representations of abstract, task-relevant quantities that could not have been previously represented[189, 191] as well as to have a causal role in task performance[34]. Further, these learned representations were stronger and more directly coupled to behavior than similar representations observed in dlPFC[192]. These results suggest that LIP may be particularly involved in highly trained and directed behavioral tasks; however, the role of LIP in undirected behavioral tasks is less clear. During passive viewing tasks in which the animal is required to maintain fixation while a sequence of stimuli are presented, LIP has been shown to be less engaged than when the same stimuli are presented in an active, directed task context[193]. In addition, inactivation[194] of LIP has been shown to bias the stimulus selected by animals in the free-choice saccade task, where the animal must select one of two identical targets toward which to saccade. However, even in these tasks, the animal's behavior is highly constrained; for instance, saccades either must be withheld, as in passive viewing, or are compelled to particular targets, as in the free-saccade choice task. In particular, it remains unknown whether LIP serves a general role in the bottom-up allocation of spatial attention in more naturalistic and undirected behavioral tasks, or if its role is specific to directed and highly constrained tasks. This lack of clarity belies a more general question about the neural underpinnings of behavior. To study complex behavior, it is necessary to use complex tasks that often require extensive training. However, it is not clear to what extent that training shapes the engagement of different brain areas and their underlying neural representations. Understanding these effects is necessary for understanding whether it is appropriate to make inferences about more natural behaviors from data collected during highly trained, less natural tasks.

Here, we compare the activity of the same populations of neurons in LIP across two distinct tasks. In both tasks, monkeys direct their gaze toward one of two presented images. However, the tasks vary in the factors which drive the animals' choice; in particular, one task is directed, the saccade delayed match-to-sample task (sDMST), and the other is undirected,

the preferential looking task (PLT). The sDMST is highly trained and requires the animal to make a saccade to a previously shown target image. The PLT requires almost no training and allows the animal to freely view the two images, which vary in their degree of familiarity to the animal. In particular, each image is drawn from either a set of novel images or a set of familiar images. The novel images are shown around ten times in a single experimental session. In contrast, the familiar images have been seen by the animal at least one thousand times in previous sessions. The same sets of images are shown in both tasks, and they are presented in the same locations. If our understanding of LIP from directed task contexts generalizes to undirected task contexts, we would expect to find that LIP is preferentially involved in the PLT, due to its reliance on innate features of the stimulus that capture attention in a bottom-up way, rather than on explicit top-down search as in the sDMST. Instead, we find that LIP has both earlier and stronger representations of both the animal's saccade choice and the image's relevance in the sDMST relative to the PLT. These results indicate that LIP is preferentially involved in directed relative to undirected tasks. Further, they indicate that representations in LIP are strongly shaped by task training and that the role of LIP in undirected and untrained tasks cannot necessarily be inferred from highly trained and directed tasks.

4.2 Results

4.2.1 Undirected and directed tasks with the same stimuli and behavioral response

To probe differences in engagement of LIP during undirected and directed behaviors, we developed two tasks that involved the same physical action (a saccade to an image target), but relied on entirely different motivation structures and training histories. The first task is untrained and allows the animal to freely view two images that are drawn from sets of novel

and familiar natural images. In this task, a fluid reward is provided at the end of the free viewing period as long as the animal correctly initiates the trial. This task is referred to as the preferential looking task (PLT, Figure 4.1b). The second task is trained and requires the animals to make a saccade to an image that matches a previously shown target image to receive a reward; the images used here are the same as those used in the PLT and they are placed in the same spatial positions. This task is referred to as the saccade delayed match to sample task (sDMST, Figure 4.1d). Importantly, in the moment just after the fixation point disappears, the animal is performing the same physical action in both tasks: They make an eye movement to one of the two presented images.

In the undirected preferential looking task (PLT, Figure 4.1b), each trial begins with a 500 ms fixation period. After the fixation period, the fixation point disappears and two images appear, in the same locations as in the sDMST. The images themselves are drawn from several natural image databases (see *Natural image sets* in *Methods*) the same image sets as in the sDMST. After 2.5 s of free viewing, the animal is provided with fluid reward. This reward is only contingent on completion of the initial fixation period, not on behavior during the free viewing period. The animal's first saccade in the free viewing period goes to one of the two images on almost every trial (Monkey R: 97 %; Monkey N: 96 %). Further, we manipulated the animal's level of interest in each image by repeatedly showing a subset of the images to the animal at least one thousand times prior to recording using a dimming detection, passive viewing task (see *Dimming detection task* in *Methods*). These previously shown images are referred to as the familiar image set. A second set of images were replaced for each recording session – and, thus, are referred to as the novel image set. This manipulation of image salience also affected behavior, as the animal's first saccade was significantly more likely to go to either a novel (95 % confidence interval for Monkey R: 52 % to 55 %; all intervals reported are bootstrapped 95 % confidence intervals unless otherwise noted) or a familiar image (for Monkey N: 61 % to 65 %; see Figure 4.1c for these biases on individual sessions).

Further, while this first saccade bias is somewhat modest, both animals spent over 30 % more time viewing images from their preferred set in the first 100 ms to 350 ms of the free viewing period than they spent viewing images from their dispreferred set (Monkey R: 25 % to 39 % more time on novel images across sessions; Monkey N: 24 % to 54 % more time on familiar images across sessions; see Figure 4.7b).

In addition, we trained the same two monkeys (R and N) to perform the sDMST. As briefly described above, the task begins with a fixation period, followed by the foveal presentation of a sample natural image. The images were drawn from the same two image sets as in the PLT. After the image was shown for 500 ms, it disappeared and the animal had to maintain fixation for a 1 s delay period. At the end of the delay, an array of two test images appeared, one in each hemifield with 180° of angular separation. One of the test images is always the same as the sample image. After the test images appear, the animal must saccade to the test image matching the sample image within 400 ms and then hold that fixation for at least 400 ms. If the animal completes this successfully, then they are given a liquid reward. The animals perform this task with 79 % accuracy on average across all conditions (Monkey R: 79 % to 80 %; Monkey N: 79 % to 80 %; Figure 4.1e). Here, the animal is motivated by reward and conditioned by explicit training to select the match image. Thus, the animal's behavior in this first saccade is directed, while the animal's behavior in the PLT is undirected.

The behavior of the animals was similar across these two tasks, indicating a similar level of engagement in both contexts. In particular, the latency of the animal's first saccade in the free viewing period in the PLT and the test period in the sDMST was nearly identical across the two tasks (Monkey R: 3 ms to 5 ms lower in the PLT, Monkey N: 16 ms to 20 ms higher in the PLT; Figure 4.1f). This indicates that the animal's urgency and deliberation were not markedly different during either task. Further, the mean velocity of the first saccade was slightly higher in the sDMST than the PLT for Monkey R ($32\,^{\circ}\,s^{-1}$ to $38\,^{\circ}\,s^{-1}$ faster),

while the opposite was true for Monkey N ($0.7\,^\circ\,s^{-1}$ to $7.3\,^\circ\,s^{-1}$ slower). In both animals, the differences were small and the distributions of velocity were totally overlapping (Figure 4.1g). Thus, while the two tasks induced slightly different distributions of saccade velocity from both animals, these differences were both small and not consistent across animals, indicating that any systematic differences between the saccades elicited in each task are minor. Finally, we investigated the latency to trial initiation as another proxy of animal engagement. Monkey R showed no difference in initiation latency (−3 ms to 2 ms and Monkey N showed only a small difference (11 ms to 37 ms slower for the sDMST; Figure 4.1h), again indicating a similar level of engagement with both tasks. These analyses provide evidence that the animal is similarly engaged by both of these tasks, and suggest that any differences we observe in the neural correlates of these behaviors are due primarily to the differences between the tasks themselves.

4.2.2 Single neurons are more active in the directed matching task

While the animals performed both the sDMST and PLT, we recorded from single neurons and small populations of neurons in the posterior parietal cortex (PPC) and lateral intraparietal area (LIP) to gain insight into how the engagement of these regions is modulated by directed and undirected task contexts. We recorded from a total of 516 neurons across both monkeys (Monkey R: 258; Monkey N: 258). These neurons were recorded across 37 recording sessions in Monkey R and 19 recording sessions in Monkey N. In Monkey R, 25 of the sessions used single wire electrodes (FHC). In the rest of the sessions in Monkey R and all of the sessions in Monkey N, we used 16- or 24- channel probes (Plexon). During recordings, one of the two test images in both tasks was placed within the spatial response field (RF) of as many neurons as possible recorded in that session. The spatial response field was determined through the standard memory-guided saccade task (see *Memory-guided saccade* in *Methods* for task details). Where appropriate, we have combined the data from the two monkeys, as

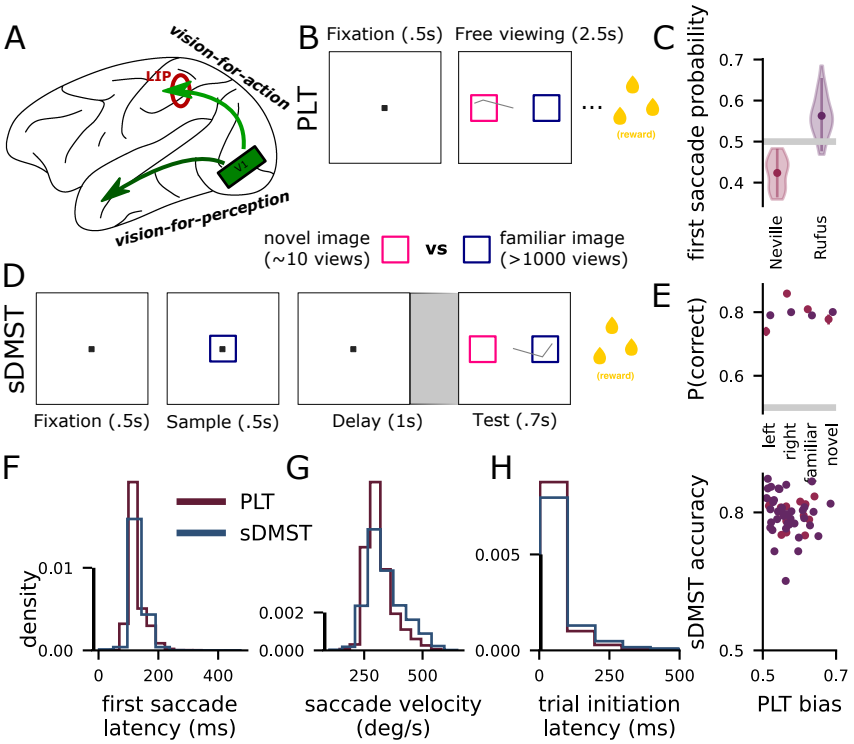

Figure 4.1: Task schematic and behavioral quantification. **A** Schematic of the Macaque visual system. Recordings were performed in the dorsal stream region LIP. **B** Schematic of the undirected preferential looking task (PLT). (below) The same novel and familiar images are presented in both tasks. **C** Violin plots of the distribution of first saccade biases in the PLT across experimental sessions. Both animals have reliable biases; Monkey R (purple) prefers novel images; Monkey N prefers familiar images. **D** Schematic of the directed saccade delayed match-to-sample task (sDMST). The images are presented in the same locations in both tasks. **E** Behavioral performance across different trial conditions. **F** Histogram of first saccade latencies. **G** Histogram of first saccade velocities. **H** Histogram of trial initiation latencies.

the results were qualitatively similar. However, in many cases, we also show separate data from both monkeys.

First, we contrasted the selectivity of these single neurons across the directed sDMST and undirected PLT. Here and in the following, we organize the data around the time of the animal's first saccade, which occurs with similar latency across both task contexts (Figure 4.1f). Using first saccade initiation as our time zero, we focus many of our analyses on the period just prior to the first saccade, which represents neural activity that could have

influenced the animal's choice (Figure 4.2a). Selectivity for task features that emerges after the first saccade could not have influenced the animal's choice. In both tasks, many neurons show elevated firing prior to the first saccade (examples: Figure 4.2a, top and bottom; population: Figure 4.2c, right) – and this firing is increased in the sDMST relative to the PLT. Further, this increased firing in the sDMST is not a consequence of the marginally increased saccadic velocity we observed in the sDMST (Figure 4.1g). To show this, we tested whether there was a reliable correlation between saccade velocity and pre-saccadic firing in both tasks across our population. This analysis did not reveal a reliable effect in either monkey (average correlation coefficient for Monkey R: $r = -0.02$ to 0 for PLT, $r = -0.01$ to 0.01 for sDMST; Monkey N: $r = -0.01$ to 0.01 for PLT, $r = -0.02$ to 0.03 for sDMST). We further tested whether this correlation was reliable across the two contexts – that is, we tested whether a neuron's firing rate-saccadic velocity correlation in the sDMST predicts the firing rate-saccadic velocity correlation in the PLT. We found no significant relationship between the correlations across these two contexts (Figure 4.2b). Thus, we find no evidence that the difference in firing rates across these two contexts are a consequence of differences in saccadic velocities, which was the most significant behavioral difference between the two contexts that we discovered, though it was inconsistent across animals.

We also compared the difference in firing rates across the two task contexts around the time the animal makes a saccade to initiate the trial (Figure 4.2c, left) with the difference in firing rates prior to the first saccade in the response period (Figure 4.2c, right). In one animal, we found that the population average difference in firing rates across the two task contexts during the trial initiation phase was much smaller than the difference during the response period (Monkey R: 0.8 z-scored difference; Figure 4.2d). This indicates that neural activity in LIP is enhanced particularly when the animal is making a directed action in the sDMST relative to an undirected action in the PLT – that is, that the response enhancement in LIP observed during the sDMST is not uniform across the task. However, there was no significant

difference between the two time periods in the second animal.

Together, these results indicate that neurons in LIP are preferentially engaged during a directed task, rather than during undirected yet active behavior. Further, we provide evidence that this enhancement of neural responses is not due to either the marginally increased saccadic velocities in the sDMST relative to the PLT or to a generalized increase in responsiveness across all task periods. Thus, this indicates that LIP may play a qualitatively different role between the directed sDMST and the undirected PLT. Next, we investigate whether these broad changes in firing rate bely differences in the population representations of important task variables.

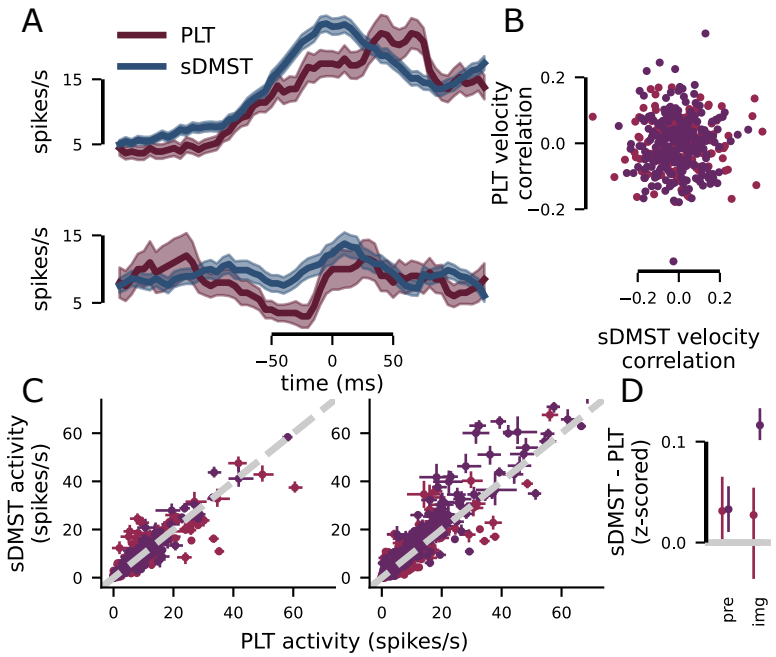

Figure 4.2: Single neurons are more engaged in the direct sDMST relative to the undirected PLT. **A** Example neurons from Monkey R (top) and Monkey N (bottom). Time is centered around the animal's first saccade in both tasks. **B** Firing rate prior to the first saccade is not reliably correlated with saccade velocity in either task. **C** (left) Firing rates across the populations of neurons in both animals just prior to trial initiation. (right) Firing rates across the populations of neurons in both animals just prior to the first saccade in the response period. **D** In one monkey, firing rates are significantly enhanced at the time of the first saccade in the response period relative to at trial initiation; in the other, there is no reliable difference. Firing rates are moderately enhanced in the sDMST relative to the PLT at both times in both monkeys.

4.2.3 Choice encoding emerges earlier in the directed matching task

The differences in responsiveness that we observe across tasks at the single neuron level suggests that we will also see changes in the information that we can decode from neural activity at the population level. First, we focus on information about the animal's saccade choice that can be extracted from the neural population. Here, the animal's physical action in both tasks is precisely the same (and happens with the same latency, Figure 4.2a). Thus, the differences we observe in saccade encoding are due to the change in motivation structure across the two tasks.

We train support vector machine (SVM) classifiers to decode saccade choice from neural activity during each of the two tasks. For all decoding analyses, training and testing trials are balanced across conditions and the decoding performance reflects performance held out trials from 10-fold cross-validation (more details of these analyses are discussed in *Support vector machine (SVM) analyses* in *Methods*). In all comparisons across the PLT and sDMST, the same neurons and the same number of trials ($N = 30$) are used. This analysis reveals that significant saccade choice decoding emerges earlier in the sDMST compared to the PLT (Figure 4.3a, left). Further, decoding performance is well above chance prior to the first saccade for the sDMST, while decoding performance rises more slowly and asymptotes much later in the PLT (from -50 ms to 0 ms relative to the first saccade, sDMST: 68 % to 89 %, PLT: 49 % to 71 %; decoding performance in the sDMST is from 2 % to 33 % higher; 284*neurons*;Figure 4.3a, right). That is, the same neural population encodes the saccade choice more strongly in the directed sDMST than in the undirected PLT. This pattern in our decoding results is replicated across individual sessions as well, where saccade decoding from individual sessions just prior to the first saccade revealed reliably higher performance in the sDMST relative to the PLT (Monkey R: 2 % to 7 % higher, Monkey N: 1 % to 7 % higher; Figure 4.3b).

Next, we ask whether neurons play similar roles in encoding saccades across these two tasks. That is, we train our SVM classifier on trials from one task and then test on trials from the other task. This cross-decoding analysis is often used to demonstrate a similar population code for quantities of interest across distinct contexts. Surprisingly, cross-decoding from both the sDMST to the PLT and from the PLT to the sDMST produces performance that is nearly identical to when the two classifiers are trained and tested on data from the same task (from -50 ms to 0 ms, trained on sDMST and tested on PLT: 51 % to 72 %; trained on PLT and tested on sDMST: 53 % to 79 %; Figure 4.3c). This indicates that the representation of saccade choice across the two tasks is very similar.

In addition, we investigated whether the strength of saccade encoding across a particular experimental session would correlate with the animal's behavioral performance on either the PLT or the sDMST. Surprisingly, we found a significant negative correlation between saccade decoder performance and the magnitude of the animal's first saccade familiarity bias on the PLT. This relationship was significant in Monkey R alone and when Monkey R and N were combined ($r = -0.83$ to -0.33; Figure 4.3c, left), but only trends toward significance in Monkey N. This indicates that LIP may become less engaged with saccade production in the undirected task as the animal's behavior is more entrained by the undirected task feature – in this case, familiarity. Next, we investigated whether a similar relationship would hold for the sDMST. While there is a weak positive correlation, it is not significant in either monkey or when the data is combined ($r = -0.20$ to 0.55, Figure 4.3c, right). However, this may be because the animal's sDMST accuracy was overall less variable from session-to-session than the animal's bias in the PLT. Further, it is significant that we did not observe the same trend in the sDMST as in the PLT (the correlation coefficients are significantly different from each other and opposite in sign). This indicates that the decrease in saccade decoding performance was not a function only of the populations recorded on each particular day.

Finally, we did observe a significant positive correlation between the animal's session-to-session looking side bias (Figure 4.7c) and pre-saccadic decoding performance on the PLT (r =0.19 to 0.77) as well as a negative correlation between the first saccade familiarity bias and this looking side bias (r =−0.76 to −0.25, Figure 4.7e). Taken together, these results suggest that LIP may become more engaged in saccade production during the PLT when the animal's behavior becomes more explicitly directed toward a particular side, rather than toward a set of images with a particular level of familiarity.

These results reveal that the neural representation of saccade choice has a core similarity across directed and undirected contexts, as indicated by the high cross-decoding performance, but also significant and surprising differences, including in latency and in the geometry of the encoding. Further, the differences in latency support a much closer and potentially causal role for LIP in the directed sDMST, but fail to provide evidence for that level of involvement in the PLT. Even beyond that, the negative correlation between decoder performance and PLT bias indicates that as the animal becomes more likely to select an image based on familiarity in the undirected task context, LIP may become even less involved in saccade production. All of this evidence together indicates that LIP may be less involved than previously thought in bottom-up, undirected behaviors.

4.2.4 Content encoding emerges earlier in the directed matching task

Our results so far depict a reduced role for LIP in undirected, yet active, behaviors, as typified in the PLT. In particular, we have shown weaker and later encoding for saccade choice. Next, we characterize the encoding of the task variables that underlie the saccade decision: The match-nonmatch status of the image in the sDMST and the familiarity of the image in the PLT. LIP has long been hypothesized to serve the role of a priority or salience map of the visual world[27]. In this context, we would then expect LIP to encode both match status in the sDMST and familiarity in the PLT.

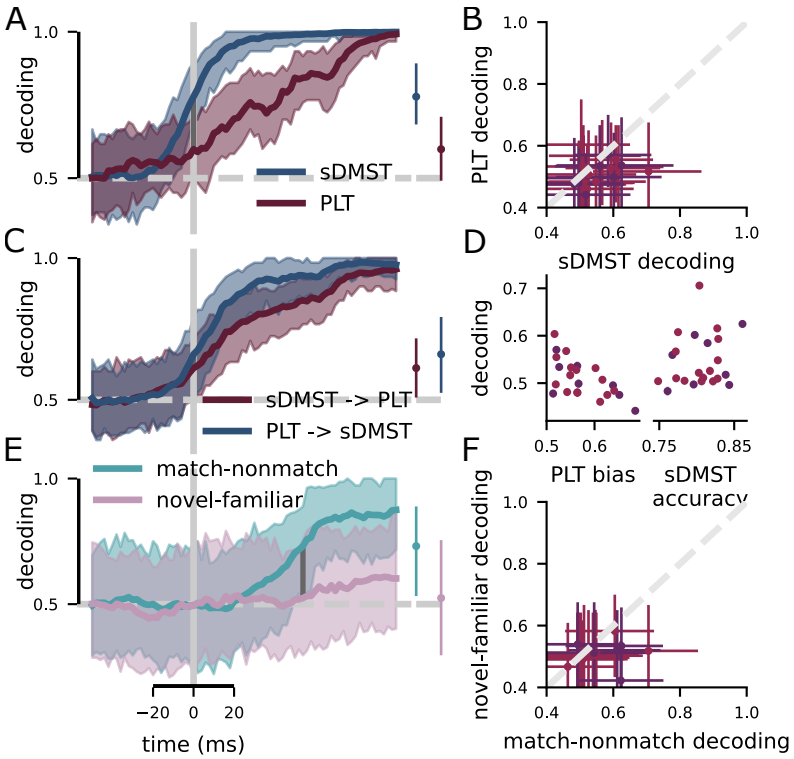

Figure 4.3: The same population of LIP neurons encodes more information about behavioral response and image relevance in the directed sDMST relative to the undirected PLT. **A** The decoding performance for SVM classifiers trained to determine whether the animal made a saccade into or out of the receptive field, contrasted across both tasks. (left) Timecourse of decoding; (right) decoding performance from −50 ms to 0 ms around the time of the first saccade. **B** Decoding performance in individual experimental sessions across both tasks prior to the first saccade. **C** (left) Saccade decoding performance in a single session is negatively correlated with looking bias in the PLT. (right) Saccade decoding performance in a single session has no clear relationship with accuracy in the sDMST. **D** Cross-task decoding performance. In the red (blue) trace, classifiers were trained on data from the sDMST (PLT) and tested on data from the PLT (sDMST). Performance is similar and high going in both directions indicating a similar representation of saccades in both tasks. (left) Timecourse of decoding; (right) decoding from −50 ms to 0 ms around the time of the first saccade. **E** Decoding performance for SVM classifiers trained to discriminate between when the image in the receptive field is either a match or nonmatch in the sDMST and novel or familiar in the PLT. (left) Timecourse of decoding; (right) decoding performance from 5 ms to 55 ms around the time of the first saccade. **F** Decoding performance just after the first saccade in individual experimental sessions.

We again use SVM decoding performance for these variables to assess the degree of encoding in the neural population. In these analyses, we balance both the training and testing sets

for the SVM classifier with equal trials in which the animal makes both behavioral responses (i.e., a saccade to each of the two images), thus marginalizing over saccade choice encoding – see *Support vector machine (SVM) analyses* in *Methods* for additional details. This analysis reveals significant decoding for match-nonmatch status that emerges after the first saccade but before the animal receives feedback about the outcome of the trial in the sDMST – however, significant decoding of image familiarity does not emerge in the same time period (from 5 ms to 55 ms after the first saccade, sDMST: 53 % to 89 %, PLT: 30 % to 75 %, difference: −10 % to 50 %, 198 neurons; Figure 4.3e). This pattern holds for individual sessions as well (Monkey R: 1 % to 14 % higher decoding in the sDMST, Monkey N: 1 % to 6 % higher decoding in the sDMST; Figure 4.4f). This indicates that LIP may primarily serve the role of a priority map in directed tasks, where there is an explicit reward contingency on the animal's choice, but less so in undirected tasks without explicit reward contingencies.

To further understand the encoding of the behavioral response and of task variables in these two contexts, we used demixed principal components analysis (dPCA)[195] to decompose neural population activity into components corresponding to each of the four outcomes for each task. In the sDMST, the outcomes are the interaction between the saccade choice (into or away from the RF) and the match status of the image in the RF (match or non-match). Similarly, in the PLT, the two saccade options interact with the familiarity of the image in the RF (novel or familiar). The dPCA analysis reveals an outcome-independent time component that is similar across both tasks (Figure 4.4a), accentuating the similarity between these two contexts. However, the analysis also reveals components related to the saccade that differ across the two task contexts. First, it reveals that a significant modulation by saccade target emerges in the sDMST before it emerges in the PLT in both animals (sDMST: −5 ms and −10 ms, Monkey R and N, respectively; PLT: 25 ms and 0 ms; $p < .01$, shuffle test; Figure 4.4b). Further, it reveals that there are two significant components of the neural response corresponding to the saccade choice for both tasks. In both tasks, one

of the components provides a stable encoding of the saccade at a particular latency, and the second component shows a reversal – in the sense that the saccade target corresponding to a positive normalized firing rate becomes the saccade target that corresponds to a negative normalized firing rate. We hypothesize that this reversing component may correspond to an encoding of the animal's next saccade. In both tasks, this saccade is often in the opposite direction of the first: In the sDMST, it is often back to the center to initiate the next trial; in the PLT, is often to the opposite image that has not yet been viewed. However, both the initial and reversed encoding emerge later in the PLT than in the sDMST. This result is also consistent with the hypothesis that the PPC uses high-dimensional representations to encode the past, present, and future behavior of an animal, as also supported by other studies of large populations of PPC neurons in the mouse[196].

Finally, the components of the response related to image relevance indicate that the encoding of match status emerges contemporaneously with the first saccade in the sDMST, while familiarity is encoded much more weakly and is significantly encoded well after the first saccade (sDMST: 10 ms and 20 ms, Monkey R and N, respectively; PLT: 85 ms and 105 ms; $p < .01$, shuffle test; Figure 4.4d, bottom). This latency of encoding is also consistent with an encoding of the animal's second saccade, as, in the PLT, this saccade significantly depends on the familiarity of the image that was viewed first. The representation of image relevance had two significant components in the sDMST and one significant component in the PLT.

Overall, these analyses further reveal that LIP is preferentially involved in directed over undirected tasks, even for its canonical function as a priority map. That is, decoding performance for image priority in the undirected context is much reduced from decoding performance for image priority in the directed context. In general, this indicates that it may be difficult to make inferences about neural activity in naturalistic contexts from neural activity in highly trained and directed experimental contexts.

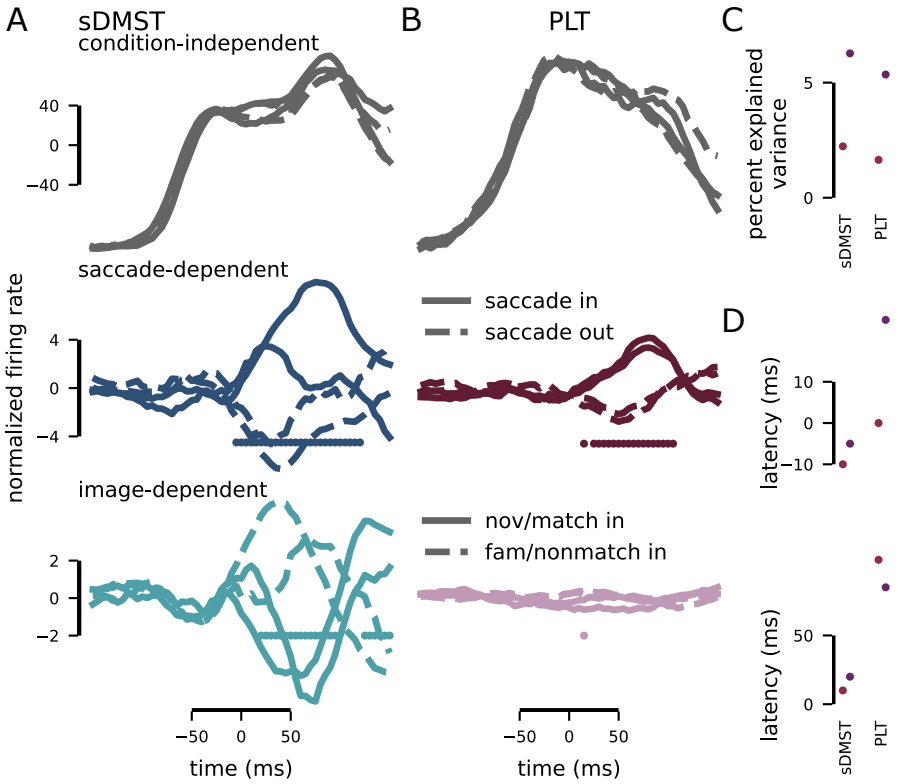

Figure 4.4: Neural population dynamics reflect greater engagement in the directed sDMST than the undirected PLT. **A** Demixed principal components analysis (dPCA) on the neurons recorded in Monkey R in the sDMST. (top) The condition independent response component; (middle) the first dimension of the saccade-dependent response component; (bottom) the first dimension of the match-dependent response component. **B** The same as C, but applied to the PLT. (bottom) The first dimension of the familiarity-dependent response component. **C** Variance explained by the dPCA models for the sDMST (left) and PLT (right), across both monkeys (colors). **D** The latency of significant modulation in the saccade-dependent (top) and match/familiarity-dependent (bottom) response components, shown for both monkeys.

4.2.5 Low-level features are also weakly encoded in the undirected task

Image familiarity is a high-level feature that relies on the animal's built up experience with the image. While signatures of image familiarity have been observed in several brain regions[197, 198] (including potentially LIP[199]), and particularly in the inferotemporal cortex[200–202], it is possible that LIP may still act as a priority map, but only for low-level features. To test this hypothesis, we included a third task in the experiments described above that was identical to the PLT except that the novel and familiar natural image sets were

replaced with high and low-luminance squares of the same size as the images and displayed in the same positions as in both other tasks(Figure 4.5a). We refer to this modified PLT as the lumPLT. The animal's behavior preference was less consistent here than in the standard PLT, however both animal's had significant biases toward the high- or low-luminance squares in 18 experimental sessions (Figure 4.5b; Monkey R: 12; Monkey N: 6). In addition, the latency of first saccades in the lumPLT were significantly longer than in the sDMST or standard PLT (Monkey R: 34 ms to 43 ms longer in the lumPLT; Monkey N: 68 ms to 86 ms longer in the lumPLT; Figure 4.5c).

Neural activity recorded during this behavior is modulated by saccade target, as with both other tasks, and single neurons show similar dynamics as in the other task contexts (Figure 4.5d). Further, there is a significant difference in firing rates between the lumPLT and sDMST (Monkey R: 56/67 neurons fire more in the sDMST; Monkey N: 51/90). The average difference in firing rates is larger in magnitude than between the sDMST and PLT (; Figure 4.5e). Thus, while the broad contrasts between the sDMST and PLT hold for contrasts between the sDMST and lumPLT as well, the animal may be significantly less engaged during the lumPLT than the PLT. Next, we performed decoding analyses on these data to contrast population encoding of saccade and of match and luminance.

Our population analyses also reveal broad similarities between the relationship between the sDMST and the lumPLT to those we have already established between the sDMST and the standard PLT. In particular, our decoding analysis (16 trials per condition) reveals that saccade encoding emerges significantly later in the lumPLT than the sDMST (Figure 4.6a, left, 143 neurons). Further, significant decoding performance emerges just after the first saccade in these data for the sDMST (from -40 ms to 10 ms around the first saccade, sDMST: 54 % to 81 %, PLT: 43 % to 70 %, difference: -9 % to 29 %, 99 neurons; Figure 4.6a, right), but does not emerge until 70 ms after the first saccade in the lumPLT. This pattern is similar

across individual recording session as well (−2 % to 5 % higher decoding performance in the sDMST, Figure 4.6b). For encoding of match status and luminance, the results are less clear across the combined population (88 neurons, Figure 4.6c, d). However, trends are similar to those in the comparison between the sDMST and PLT from before.

Thus, these results suggest that LIP does not preferentially represent only low-level features in a priority map, but rather is still preferentially engaged by directed tasks relative to undirected, yet active, behaviors. This runs counter to narratives that suggest LIP provides a priority map for guiding the deployment of spatial attention based on low-level features, like luminance and spatial frequency. Instead, this work suggests that LIP is preferentially involved in directed and extensively trained behaviors with explicit reward contingencies. While we still believe that there is likely to be a spatial priority map represented somewhere in the brain, our results suggest that LIP is not the locus of that map.

4.3 Discussion

Taken together, these results indicate that LIP has a preferential involvement in highly-trained, directed behavioral tasks compared to more naturalistic and undirected behaviors. LIP appears to be less involved in guiding saccades in the undirected PLT than in the directed sDMST, even when the behavioral response, and the target of that response, is the same across both tasks. Similarly, LIP appears less involved in the representation of image relevance in the undirected than in the directed task. This pattern of results is replicated on an even simpler undirected free viewing task where the animal chooses to view either a high- or low-contrast square. Thus, the lack of engagement of LIP during the undirected task is not due to our choice of familiarity as the manipulated image feature in the main undirected task. These results indicate that LIP may be more extensively shaped by training and explicit task demands than previously believed – and may, in fact, be preferentially engaged only in tasks

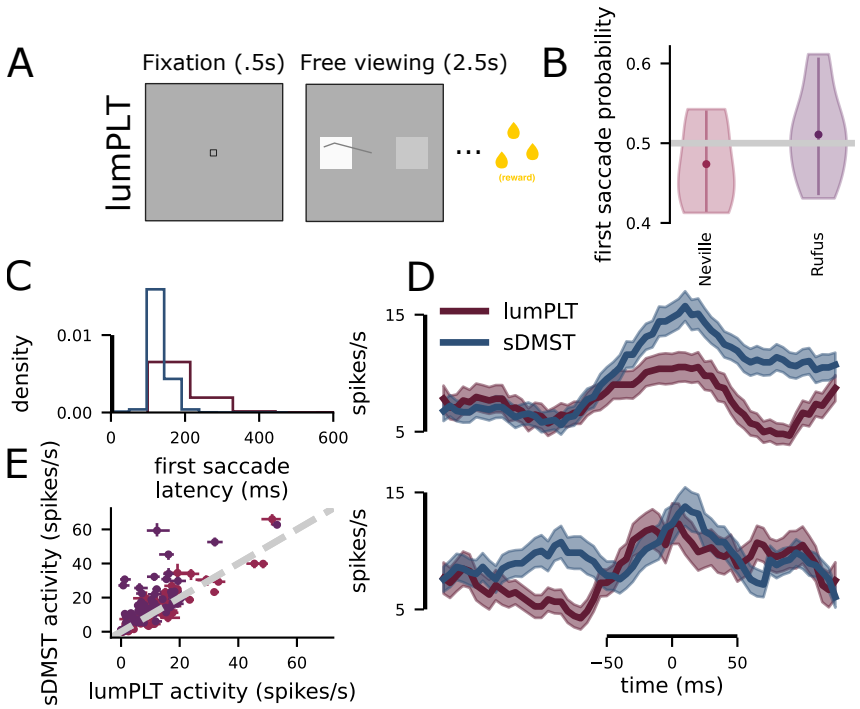

Figure 4.5: An undirected task where behavior is entrained by a low-level stimulus feature. **A** A variant of the PLT, which is referred to as the luminance preferential looking task (lumPLT). It is the same as the PLT except that luminance squares with either high- or low-contrast are presented instead of natural images. **B** On different sessions, the animals have different biases for viewing either low (left) or high (right) luminance squares. **C** The latency of the animal's first saccade is slightly greater in the lumPLT compared to the sDMST. **D** Example neurons from Monkey R (top) and Monkey N (bottom); time is organized around the animal's first saccade. **E** Neurons have significantly higher firing rates prior to the first saccade in the sDMST relative to the lumPLT.

that have explicit direction, while undirected, but similar, tasks may be mediated by other brain regions, such as FEF.

We believe that these results cast doubt on a common form of inference in neuroscience. In particular, many experimental tasks are designed to, in part, mimic naturalistic behaviors. For instance, search tasks are often analogized to navigating in cluttered environments with many potential stimuli to select from. Then, neural signals are recorded in the highly trained task with explicit reward contingencies, and used to infer the roles of different brain regions in the more naturalistic behaviors that may not have immediate or explicit reward

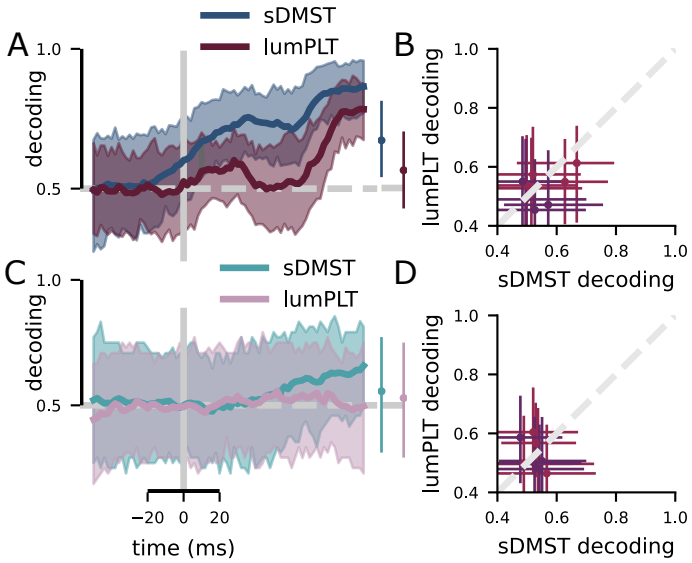

Figure 4.6: LIP is also less involved in undirected behaviors that depend on low-level stimulus features. **A** The decoding performance of SVM classifiers trained to determine whether the animal made a saccade into or out of the receptive field (as in Figure 4.3a). (left) The timecourse of decoding; (right) decoding performance −40 ms to 10 ms around the time of the first saccade. The pattern is similar to that shown for the PLT and sDMST. **B** Decoding performance for saccades in individual experimental sessions; similar to Figure 4.3b. **C** The decoding performance of SVM classifiers trained to determine whether there was a match or nonmatch and high- or low-contrast square in the receptive field; similar to Figure 4.3a. (left) The timecourse of decoding; (right) decoding performance 55 ms after the first saccade. **D** Decoding performance for image relevance in individual experimental sessions; similar to Figure 4.4b.

contingencies. Our results cast doubt on that inference step by showing that extensive training and explicit reward structure can fundamentally alter the role of different brain regions in similar behaviors, with the same behavioral response. These results emphasize the need to study naturalistic behaviors in parallel with highly trained and well-controlled tasks to gain a more generalized understanding how neural function depends on behavioral context.

Our results may be surprising in the context of recent results showing that inactivation of PPC, including LIP, significantly affects how correlated an animal's free viewing behavior is with the low-level salience map of a natural image[203]. Those results indicate that LIP

has a role in guiding the deployment of visual attention according to the low-level visual salience of an image, outside of a directed task context. First, our main free viewing task did not rely on the same kind of low-level visual salience – instead, we used a memory-related image familiarity signal to sculpt the animal's behavior. Thus, it is possible that PPC is more involved in the representation of low-level image salience than image familiarity, which is more abstract. However, if this were the case, we would expect our high- and low-contrast control task to have revealed that role, though it remains possible that familiarity is too complex and contrast changes are too simple to produce a higher level of engagement from LIP. A second difference that could explain our conflicting results is in the presentation of the images. In the inactivation study[203], the images covered much of the visual field and the animal made saccades around the same natural image for the duration of a free viewing trial. It is possible that PPC has a specialized role in representing the relative salience of nearby spatial locations, consistent with its previously described role in spatial processing[18, 19] and its described role in pop-out visual search tasks[99]. However, in our experiments, we also included trials on which the two stimuli were presented with only 45° of separation in visual angle in both the PLT and the lumPLT. We did not see a significant difference in either familiarity or contrast decoding between this condition and the 180° separation condition (Figure 4.9). While this does not precisely replicate pop-out conditions, it indicates that closer spatial arrangement alone is not enough to engage LIP in undirected free viewing tasks to the same degree as it is engaged in directed search tasks.

In light of these results, there are two alternative ways to reconcile the conflicting conclusions from the two studies. First, the disruption to salience-related behavior may be due to a generalized disruption of visual – and potentially spatial[18, 19] – processing that results from the inactivation of PPC. Second, the effect on salience preference reported in the inactivation study[203] is relatively small, and indicates that LIP may not be the primary region involved in producing that salience preference. Thus, LIP may play a minor role in

producing salience-related behavior that our experiment here did not detect. However, if this is true, our conclusion that LIP is more involved in representing image relevance in a directed rather than undirected context still holds.

This work illustrates the importance of studying neural function in a variety of both naturalistic and highly-controlled behavioral contexts. In this case, additional work must be performed to further isolate the role of LIP in diverse and complex behaviors. Together with previous work, our results highlight the malleability of LIP during task training and suggest that LIP may be preferentially involved in directed tasks with strict behavioral contingencies. This new conceptualization of LIP makes an interesting prediction: Our results predict that LIP would not be involved in search tasks with pop-out effects in general, but only when stimulus selection is enforced by a direct task. For free viewing of pop-out arrays, in which the pop-out stimulus is still likely to capture attention, we would expect a reduced role for LIP given our findings here.

Acknowledgments: This work was supported by NIH F31EY029155 (WJJ),
NIH R01EY019041 (DJF), CRCNS NIH R01MH115555 (DJF), NSF NCS 1631571 (DJF), and a DOD Vannevar Bush Fellowship (DJF).

Author contributions: WJJ and DJF conceived of the project and developed the experiments. WJJ and SMT trained the animals and performed the electrophysiological recordings. DJF supervised the experimental work. WJJ analyzed the data and made the figures. WJJ and DJF wrote the paper.

Competing interests: The authors declare no competing interests.

4.4 Methods

4.4.1 Surgical preparation and experimental setup

Two male monkeys (Macaca mulatta; Monkey R: 10 years old, ∼10 kg; Monkey N: 14 years old, ∼14 kg) were implanted with head posts and then trained on four behavioral tasks described below. Monkey R was implanted with a recording chamber positioned over PPC after training was complete. Monkey N had a recording chamber positioned over LIP from a previous series of motion categorization experiments that was used in this set of experiments as well. He had the chamber during training and it was maintained with regular sterile cleaning throughout. Our surgical, behavioral, and neurophysiological approach has been described in detail previously[135, 204]. Stereotaxic coordinates for chambers placement were determined using structural magnetic resonance imaging (MRI) that was performed prior to the headpost implantation to avoid artifacts. Both chambers were centered on the intraparietal sulcus. In Monkey R, the chamber was centered over the right cortical hemisphere, 23.8 mm lateral from the midline and 1 mm posterior to the intra-aural line. In Monkey N, the chamber was centered over the right cortical hemisphere, 9 mm lateral from the midline and 5.25 mm posterior to the intra-aural line. Both monkeys were housed in individual cages with a 12 hour light-dark cycle. Behavioral training and recordings always occurred during the light part of the cycle. During behavioral training and recordings, monkeys sat in custom-made primate chairs and were head-fixed. Task stimuli were displayed on a 21-inch color CRT monitor (1280*1024 resolution, 75 Hz refresh rate, 57 cm viewing distance). Identical timing and rewards were used for both monkeys. Different image sets were used for across the different monkeys and the details of this are described below, in *Natural image sets* in *Methods*. A solenoid-operated reward system was used to deliver juice reward to the monkeys. Monkeys' eye positions were monitored by an optical eye tracker (SR Research) at a sampling rate of 1 kHz and stored for offline analysis. Stimulus presentation, task events, re-

wards, and behavioral data acquisition were accomplished using an Intel-based PC equipped with MonkeyLogic software running in MATLAB (http://www.monkeylogic.net)[205, 206]. All experimental and surgical procedures were in accordance with the University of Chicago Animal Care and Use Committee and National Institutes of Health guidelines.

4.4.2 Behavioral tasks and training

Our two main tasks (the sDMST and PLT) are described in detail in the main text. Some additional detail is given here. Two additional tasks that were used in the animal's training are described in detail here as well. The memory-guided saccade task was used to functionally probe for LIP-like responses[207, 208] and the dimming detection task was used to familiarize both animals with their familiar image sets.

Both animals were trained to perform the sDMST over six months to one year. First, both animals were trained on the task using solid color squares (red and green). Then, a reduced set of natural images was introduced and both animals were further trained. Finally, both animals were trained on the task with the set of familiar images used in the experiments and different sets of novel images until they reached consistent levels of performance on those image sets as well. In both animals, near the end of their training on the sDMST, they were trained on the dimming detection task, which was used to familiarize the animals with their familiar image set.

Memory-guided saccade

The memory-guided saccade task is used to establish whether recordings are made from LIP, as neurons in LIP are believed to exhibit characteristic response properties in this task. In the task, the animal initiates a trial with a fixation period. Then, a small luminance target is flashed for 300 ms at one of eight locations, equally spaced in the periphery. After a 1000 ms

delay period in which the animal must hold fixation, the fixation point disappears and the animal must make an eye movement to the location of the target flash. There is no target present on the screen at the time of the saccade. If the animal completes the saccade within 500 ms and holds fixation for 150 ms, then they receive reward.

LIP neurons are believed to show some combination of spatially selective visual, delay, or presaccadic activity during this task. However, not all neurons in LIP show these properties.

Dimming detection task

This task was used to familiarize both animals with their familiar image set, by repeatedly presenting images from the familiar image set while maintaining the animal's engagement.

The task begins with a fixation period, then a sequence of images is presented in the fovea. The animal is required to maintain fixation throughout the sequence. Each image is presented for 450 ms. On half the trials, there is an equal chance that 1 through 5 images will be presented. For all of these sequence lengths, the last image will dim and the animal has 450 ms to indicate that they perceived the dimming by releasing a touch bar to receive a reward. On the other half of trials, a sequence of 6 images is presented and none of them dim. To receive reward, the animal must hold the bar through the entire sequence.

4.4.3 Natural image sets

Both animals had a single set of familiar images that remained constant throughout the experiments. In Monkey R, this set consisted of 55 images, but 40 images were randomly selected for use in each experimental session. In Monkey N, this set consisted of 40 images and the whole set was used in each experimental session. All image sets (both novel and familiar) for each experimental session across both monkeys was randomly selected from the Corel Image Library. The images were cropped to be 150x150 pixels. These images

were not controlled for their low-level features or for their contrast, as doing so often made them appear unnatural and we believe that this would introduce a bigger confound than attempting to control for these properties. We evaluated the effect of low-level image salience on animal viewing behavior in the PLT and did not find a reliable relationship (see *The effect of low-level salience on behavior* in *Supplement*).

4.4.4 Electrophysiological recording

In Monkey R, neuronal activity was recorded using 75 µm tungsten microelectrodes (FHC) or 16-channel V-Probes (Plexon). In Monkey N, neuronal recordings were made using either 16- or 24-channel V-Probes (Plexon). All 16-channel probes had 100 µm spacing between electrode sites; 24-channel probes had 75 µm spacing. In both cases, all of the sites were arranged in a line. In Monkey R, 25 of the sessions used single wire electrodes (FHC). In the rest of the sessions in Monkey R and all of the sessions in Monkey N, we used 16- or 24-channel probes (Plexon). During recordings, one of the two test images in both tasks was placed within the spatial response field (RF) of as many neurons as possible recorded in that session. The spatial response field was determined through the standard memory-guided saccade task (see *Memory-guided saccade* in *Methods* for task details).Neurophysiological signals from both single or multi-channel recording were amplified, digitized and stored for offline spike sorting. In both monkeys, we recorded neuronal activity in the memory-guided saccade task to map LIP RF locations.

We localized LIP in each monkey according to the pattern of neuronal activity during the memory-guided saccade task, as described above. In both monkeys, we also considered neurons recorded from the same grid holes and at similar depths as previous recordings that yielded LIP-like memory-guided saccade activity to also be in LIP. In all recordings, we also identified LIP neurons based on anatomical criteria, such as the location of each electrode track relative to that expected from the MRI scans, the pattern of gray–white

matter transitions encountered on each electrode penetration, and the relative depths of each neuron.

Spike sorting

In Monkey R, 44/66 of the datasets were sorted offline by hand (Plexon) to verify the quality and stability of neuronal isolations. In the remaining 22/66 of the datasets from Monkey R and for all of the data from Monkey N, single neurons were sorted automatically using Kilosort 2 (code available here)[209]. All analyses were performed on neurons marked "good" by that software, which indicates that less than 10 % of recorded spikes are likely to have arisen from a different neuron. Postprocessing of these sorted neurons was performed in Phy (code available here).

Postprocessing

For all analyses reported here, the data from each neuron was smoothed with a 50 ms wide causal boxcar filter. That is, the spike rate at time zero includes all of the spikes from −50 ms to 0 ms.

Where noted, we also z-scored these firing rates. In every case, z-scoring was performed within the frame of reference of the analysis; that is, the data used for a support vector machine classifier analysis was all z-scored before being broken into different classes and training and test sets.

4.4.5 Data analysis

Exclusion criteria

Only sessions in which monkeys performed enough trials for each unique stimulus condition in both the sDMST and PLT tasks ($n \geq 30$ for monkeys R and N) were used for further

analysis, except in Figure 4.6 where neurons with $n \geq 24$ trials and 10 spikes per trial were included. Further, in pseudo-population analyses, only trials in which the neuron fired more than 5 spikes through the whole trial were included. In addition, the neuron totals for each pseudo-population analysis are reported in the main text. In addition, for population analyses, only recordings with more than five neurons are included.

Saccade detection and analysis

To detect saccades, we first applied a median filter to the recorded eye traces to remove noise. Then, saccades were defined as periods within the smoothed eye trace during which the instantaneous velocity of the eye was greater than $70\,^{\circ}\,\mathrm{s}^{-1}$. Any fixations that lasted less than 10 ms were assumed to be artifactual and therefore removed so that the previous and succeeding saccades were merged into a single saccade. Velocity for the whole saccade was computed by taking the distance between the eye position at the beginning and end of the saccade, divided by the duration of the saccade.

We organize our analyses around the time of the animal's first saccade in the response period (that is, after the fixation point disappears in both tasks, and the free viewing array appears in the PLT and the test array appears in the sDMST). Time zero in all of the plots with time on the x-axis is the time of the initiation of that saccade.

Support vector machine (SVM) analyses

We trained support vector machine (SVM) classifiers to decode both the animal's saccade choice (i.e., a saccade into or away from the RF of the recorded neurons) and information about the image in the RF. In all decoding analyses the number of trials in each condition were balanced. Further, when decoding information about the image in the RF, an even number of trials where the animal made a saccade into and away from the RF were included

– thus, marginalizing over the animal's motor action. These balanced sets of trials were resampled 100 times to account for the variability that is introduced by choosing only a subset of trials. These 100 resamplings were used to produce 95 % confidence intervals for our SVM plots, as displayed.

All of our SVM analyses depend on the SVM implementation provided in scikit-learn[146]. We used the radial basis function kernel in all cases except for the cross-decoding analysis (where we used a linear kernel). Performance was not qualitatively different when using a linear kernel for all the analyses.

Demixed principal components analysis (dPCA)

For the demixed principal components analysis (dPCA)[195], we used a modified version of the implementation provided by the Machens lab here. The modified version is available here. This modified version incorporates two bug fixes; we provided one and one is provided by Thijs van der Plas. Neither change any of the theoretical underpinnings of dPCA.

In our case, dPCA was fit to a pseudopopulation across all of our experimental sessions, separately for each monkey.

Data and code availability

All code used to generate the figures and perform the analyses is written in python and available here. This code relies on additional custom code available here. Finally, it also makes extensive use of the python scientific computing environment.

4.5 Supplement

4.5.1 Additional behavioral quantification in the PLT

Here, we describe an additional way of evaluating the animal's familiarity bias as well as investigate the animal's image-independent bias toward a particular side. As reported in the main text, on most experimental sessions both animals exhibited a significant familiarity bias in their first saccade (Figure 4.7a). Another way of looking at this evaluates the amount of time that the animal spends viewing the novel relative to the familiar image in the early part of the PLT free viewing period. That is, we take the 100 ms to 350 ms after the onset of the free viewing period, which encapsulates most of the animal's first and second fixation. Then, we take the amount of time their eyes were on the familiar image and subtract that by the amount of time their eyes were on the novel image. Then, we divide by the period length (250 ms). This gives us a looking time familiarity bias that ranges from -1 to 1, where -1 indicates that the animal spent all of their time viewing the familiar image and 1 indicates that they spent all of their time viewing the novel image. Both animals have significant, and quite large, familiarity biases according to this metric as well (Monkey R: 0.26 to 0.39; Monkey N: -0.52 to -0.24; Figure 4.7b).

Finally, we also compute the animal's side bias for their first saccade. That is, we ask if the animal is more likely to look to the image in one hemifield independent of any of that image's properties. Both animal's have often significant side biases that vary across sessions (mean side bias from different experimental sessions, Monkey R: 0.43 to 0.54, Monkey N: 0.26 to 0.4; Figure 4.7c). Though in many behavioral sessions, this bias is relatively small.

Next, we ask about the interactions between these three quantities. First, we ask whether the animal's first saccade bias and looking time bias are correlated with each other across sessions. That is, when the animal has a strong first saccade bias, do they also have a strong

looking time bias? Interestingly, we do not find a reliable relationship between these two quantities (Monkey R: $r = -0.54$ to 0.06, Monkey N: $r = -0.48$ to 0.47; Figure 4.7d). Next, we asked if the strength of the animal's side bias predicted the strength of the animal's first saccade bias. Here, we do find a strong negative relationship in both monkeys (Monkey R: $r = -0.73$ to -0.3, Monkey N: $r = -0.86$ to -0.36; Figure 4.7e). Together, these results indicate that the animal's side bias may influence that strength of the animal's first saccade bias. However, the strength of the first saccade bias is not strongly related to the animal's overall familiarity preference.

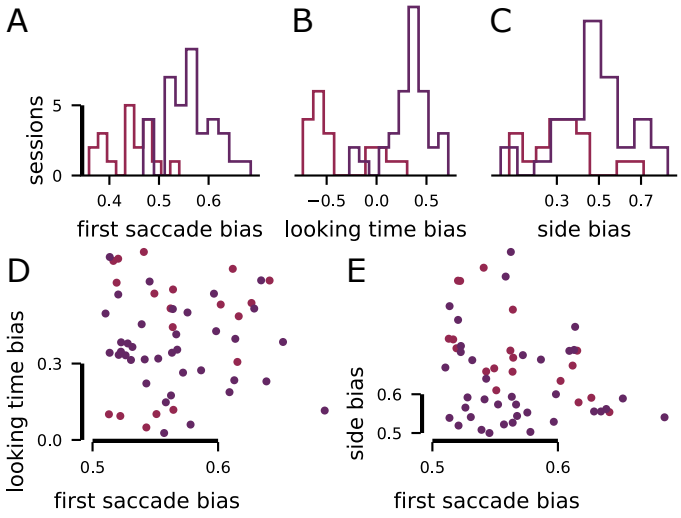

Figure 4.7: Additional quantification of the animal's behavior on the PLT, related to Figure 4.1. **A** Histogram of the first saccade novelty bias across different experimental sessions. **B** Histogram of the looking time novelty bias across different experimental sessions. **C** Histogram of the side bias across different experimental sessions. Here, 1 means the animal always looked into the recorded hemifield, and 0 means the animal always looked away from the recorded hemifield. **D** Scatter plot between the first saccade bias and looking time bias. This plot indicates no significant relationship between these two quantities. **E** Scatter plot between the first saccade bias and the side bias. This plot reveals that, as the side bias becomes larger, the first saccade bias tends to decrease.

4.5.2 The effect of low-level salience on behavior

One possibility is that the animal's behavior in the PLT is strongly entrained by differences in the low-level salience of the two presented images. Low-level salience is derived from areas

within each image of high contrast or high spatial frequency relative to the surroundings. Many algorithms have been proposed to compute this low-level salience from image inputs, and are benchmarked against human free viewing data[210]. One successful yet simple algorithm is Boolean map salience (BMS)[211]. BMS leverages both local salience cues like those described above, as well as some global cues that have also been shown to be used in figure-ground segregation such as surroundedness.

We evaluated whether the animal's behavior was significantly modulated by differences in image salience. To do this, we computed the low-level salience map of each of the images in the familiar image set, using the BMS algorithm, and tested whether the animal was more likely to look at the image with higher mean salience either with its first saccade (Figure 4.8a) or for more time in the initial part of the free-viewing period (Figure 4.8b) – both of which are time periods where the animal's behavior shows significant modulation by differences in familiarity. For both ways of computing salience bias, neither animal showed a reliable bias toward the more salient images across different experimental sessions. Further, the biases that were expressed in each session were almost always smaller in magnitude than the familiarity bias and likely reflect random fluctuations in the animal's behavior rather than a true salience bias (looking time bias, Monkey R: -0.04 to 0.02 difference in looking time; first saccade bias, Monkey R: 0.47 to 0.52 probability of looking to the more salient image).

4.5.3 Undirected tasks with stimuli in close spatial proximity

In addition to the conditions of the tasks focused on in the main text, we included conditions in both the PLT and lumPLT in which the two images were placed with 45° rather than 180° angular separation. This condition allows us to ask if LIP's representation of either the behavioral response or of image relevance changes when the two images are placed in close proximity to each other. As low-level salience and other methods of capturing bottom-up

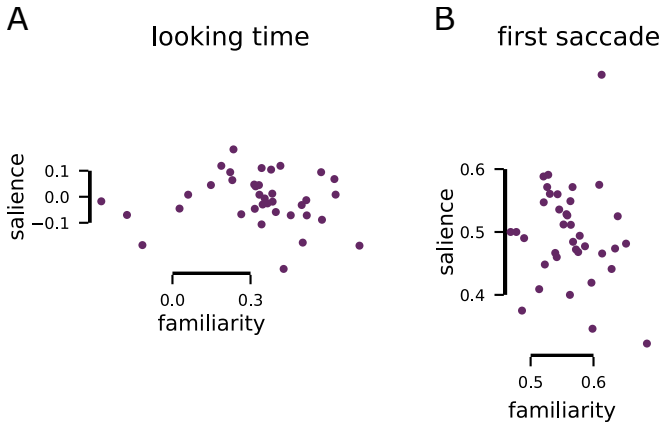

Figure 4.8: Quantification of the animal's behavior as it relates to the low-level salience of the presented images, related to Figure 4.5. **A** Scatter plot showing the relationship between the strength of the biasing of the first saccade due to differences in low-level salience or differences in familiarity. There is no significant relationship. **B** Scatter plot showing the relationship between the strength of the looking time bias due to differences in low-level saliences and differences in familiarity. There is no significant relationship.

attention are often defined relative to immediately surrounding stimuli, this closer stimulus proximity in our task might reveal a more direct role for LIP in undirected behavior.

As in the main text, we train SVM classifiers to decode both the saccade choice (Figure 4.9a,b) and image relevance (Figure 4.9c,d) when the images are placed close and far from each other. We perform this analysis on both the standard PLT (Figure 4.9a,c) and the luminance PLT (Figure 4.9b,d). In all cases, the results are similar between the close and far conditions, indicating that LIP does not play a special role for undirected stimulus selection among nearby rather than far apart stimuli.

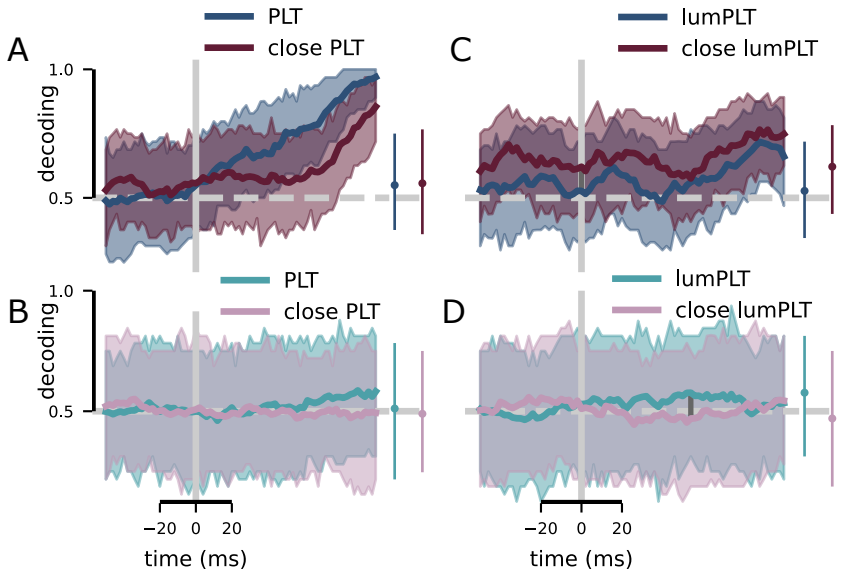

Figure 4.9: Comparison of decoding performance when stimuli are in close proximity relative to when they are in opposite hemifields, related to Figure 4.6. **A** Decoding performance for saccade choice in the standard PLT where stimuli were placed either in opposite hemifields or nearby. **B** Decoding performance for saccade choice in the luminance PLT where stimuli were placed either in opposite hemifields or nearby. **C** Decoding performance for familiarity in the standard PLT where stimuli were placed either in opposite hemifields or nearby. **D** Decoding performance for contrast level in the luminance PLT where stimuli were placed either in opposite hemifields or nearby.

CHAPTER 5

DISCUSSION

5.1 Summary of results

The previously described work – from three different perspectives – all pushes toward the same goal: Developing a unified understanding of the brain as a distributed computational system. In chapter 2, we showed that nonlinearly mixed codes for sensory, cognitive, and motor features – which have been observed across the brain – satisfy the dual goal of reliable and efficient coding, in addition to their previously demonstrated benefits for the accessible encoding of information. We outlined several additional benefits of mixed codes as well. First, we showed that they retain their benefits for reliability even when accounting for the increased population sizes necessary to implement them. We also demonstrated a trade-off between error magnitude and error probability that can be navigated by changes to receptive field size. Finally, we demonstrated that these mixed codes exist for features that do not require an accessible encoding. This indicates that the brain may exploit these codes for their reliability and efficiency benefits alone, rather than only for their benefits to accessibility.

Next, our focus increased in scale and – assuming a reliable and efficient code similar to the mixed codes of the previous chapter – in chapter 3, we demonstrated a solution to the representation assignment problem across two distinct brain regions. We showed that by leveraging information that was common between both regions, the brain can reliably integrate information from distinct sources, or that is uniquely computed only in a single brain region. This work demonstrates a reliable solution to the assignment problem, which is critical to coherent behavior in rich, naturalistic sensory environments. We then illustrated a trade-off between catastrophic assignment assignment errors and the magnitude of local errors in decoding. That is, extremely reliable assignment requires increased redundancy

between the two regions, which limits the amount of information available for representing the unique information. Finally, we showed that our framework is quantitatively consistent with human psychophysical data, and that a crucial prediction about how assignment errors depend on stimulus proximity is qualitatively satisfied.

Finally, in chapter 4, we described experimental work that probes the context-dependence of representations in LIP. There, we used two tasks: One was a directed matching task with explicit requirements on the animal's behavior; and the other was an undirected free viewing task without explicit requirements on behavior. This work revealed that LIP was preferentially engaged by the directed relative to the undirected task. This is surprising given previous work that suggests LIP plays a general role in the allocation of bottom-up spatial attention and, thus, would be expected to be involved in the undirected task. This work illustrates the need to consider behavioral context when developing an understanding of the function of a brain region.

There are many ways to build on these results. In the following, we discuss several future research directions, which either build on the results described previously, or work to integrate them more closely together. In both cases, we continue to consider ways in which we can gain a more detailed understanding of the brain as a distributed system.

5.2 Toward optimally reliable neural codes

While the nonlinearly mixed codes described in chapter 2 have numerous desirable properties as described for the first time both in that work and elsewhere[62, 63], they do not come close to approaching theoretical bounds on the maximum amount of information that can be reliably transmitted in a noisy system[61]. That is, they fail to approach bounds on channel capacity established in information theory. In particular, as the level of mixing in a code is increased, the code becomes more reliable and efficient; however, the increased

mixing increases the number of neurons required to implement the code and those additional neurons increase the theoretical capacity of the code beyond what is used in practice. Here, we consider a family of codes that can be theoretically related to the mixed codes described previously, but that have also been shown to approach theoretical bounds on reliability.

To motivate these codes, we consider a single neuron in cortex. In many studies of neural coding, including our own, the neuron is modeled in the following way,

$$r = f(x) + \epsilon \tag{5.1}$$

where r is the response of the neuron in spikes per second, x is a stimulus or stimulus vector, f is the response function of the neuron which operates on the stimulus domain (i.e., the receptive field of the neuron), and ϵ is a noise term, which is often assumed to be Gaussian distributed. In studying code reliability, the magnitude of the noise term ϵ is often taken to be relatively large in comparison to $f(x)$. This mirrors conditions observed in the brain where a neuron's average activity and the variability in that activity have been measured to be of the same order of magnitude (that is, with a Fano factor close to 1)[150]. However, two details complicate this understanding. First, when single neurons are isolated from their inputs and stimulated, they have been shown to be far more reliable than recordings of cortical neurons suggest[212]. In addition, neurons in the periphery, that have reduced numbers of distinct inputs relative to cortical neurons, have been shown to be exquisitely reliable[213]. From these observations, we would infer that ϵ is much smaller than $f(x)$ in Eq. 5.1. Second, fluctuations in firing rates in cortex that have previously been understood as noise have been shown to, instead, reliably reflect motor actions[55, 56] and, in some cases, sensory information from other modalities[56]. Thus, when accounting for this variation, we would expect the signal-to-noise ratio of cortical neurons to be much higher than when that additional signal is viewed as noise. Thus, a more appropriate model for a cortical neuron

would be,

$$r = f(\vec{y}) + \eta$$

where $\vec{y}$ is a general state vector – which encompasses information about the animal's motor actions, history, and so on – of which our original stimulus x is only a small part. Now, η is a noise term that is small in magnitude relative to $f(\vec{y})$. That is, a single neuron's inputs are likely much higher dimensional than previously appreciated – and this high-dimensionality likely produces a significant fraction of the apparent noise in neural systems.

In these conditions, single neurons begin to resemble a construct from coding and information theory known as the multiple access channel (MAC). In a MAC, multiple users all want to send a message using the same channel. In neuroscience, these multiple users might be, for instance, different sensory modalities and motor plans. Surprisingly, work on the MAC shows that there are no gains in information capacity from forcing these distinct users to send their messages at different times or only along particular frequency bands[61]. In fact, maximal capacity can be achieved if all of the different users send their messages at once. The analog to this in neuroscience would be similar to what is observed in cortex: A single neuron at any given time should reflect a diversity of inputs, corresponding to distinct modalities and neural systems.

This surprising insight provides both a non-obvious understanding of the heterogeneity in neural responses and a potential mechanism by which neural codes can make full-use of their signaling capacity – that is, by taking a superposition of codes for many different factors of the sensory environment as well as of the motor and cognitive state. This superposition strategy has been shown to approach capacity in the coding theoretic context[214]. However, some challenges remain in seeing how it would be implemented in the brain. In particular, in many capacity achieving codes, decoding requires complex and non-local operations that

have no clear implementation in neuroscience. Thus, future work is required to understand how this superposition coding strategy can be implemented in realistic neural populations while allowing for the encoded information to be rapidly used by downstream brain regions.

5.3 Correlated neural activity and reliable integration

As alluded to in chapter 3, relatively little work has focused on characterizing how the requirement that brain regions simultaneously represent multiple stimuli affects theories of normative neural representations. Interestingly, one of the few studies that does focus on this question indicates that nonlinear mixed selectivity (of the kind discussed in chapter 2) is useful for reducing ambiguity in this context as well[215]. This is because mixed representations bundle distinct stimulus features together in representation space and thus avoid ambiguity when attempting to reconstruct multiple stimuli with multiple features – this is precisely the same assignment problem that we discuss a solution to in chapter 3, except that it takes place within single brain regions that are composed of neurons which represent only one stimulus feature (or represent several stimulus features with only linear mixing).

Further work has characterized how different ways of mixing representations of multiple stimuli affect the accuracy of a decoder for those stimuli[163]. In particular, the study compared linear mixing of the multiple representations, where the total response was a weighted sum of the responses to each of the individual stimuli, and nonlinear forms of mixing the multiple representations, such as divisive normalization. In both cases, stimuli that were nearby to each other had decreased decoding accuracy – that is, they negatively interfered with each other. However, in some cases, nonlinear mixing of the stimuli reduced that interference[163].

These results have interesting implications for the results described in chapters 2 and 3. First, they further indicate the importance of high-dimensional representations within brain

regions, even in the context of the assignment problem across brain regions. This is because, as discussed in chapter 3, high-dimensional representations will, on average, be further apart than lower-dimensional representations. Thus, the interference between nearby stimuli discussed above is less likely to occur for high-dimensional representations. Further, the analysis in [163] also reveals an effect that gains significance in the context of representation assignment across regions: For a broad family of noise correlation structures, as stimuli come closer together, the estimates of the stimuli become strongly negatively correlated with each other[163]. This negative correlation has two immediate consequences: First, it reduces the accuracy of estimates of the distance between stimuli – which are likely to be extremely important to behavior; second, it could lead to higher rates of assignment errors in some conditions, as negatively correlated estimates are more likely to produce the swaps in representation position that lead to an assignment error.

In both cases, positively correlated stimulus estimates would be preferred. As an example, a correlation coefficient of 1 would mean that estimates of the distance between the stimuli remain perfectly accurate despite inaccurate estimates of the stimuli themselves. Similarly, the probability of an assignment error would be zero. Interestingly, positively correlated stimulus estimates within each region could provide a mechanism by which the assignment error rate is reduced without actually increasing the redundancy between the two regions. However, further work is necessary to understand how these positive correlations can be achieved in practice, and whether the correlation structure that they require between neurons is feasible given what is understood about the structure of noise correlations and how those correlations change for representations of multiple nearby stimuli.

5.4 Elaborating the function of dorsal and ventral visual regions in undirected tasks

While the experimental work described previously is focused on the lateral intraparietal area (LIP) in the posterior parietal cortex (PPC), which is located in the dorsal visual stream, we also performed a related experiment while recording from small populations of neurons in the inferotemporal cortex (ITC), which is in the ventral visual stream. In these experiments, the animal performed the preferential looking task (PLT) described in chapter 4. Briefly, the monkey initiated each trial with a 500 ms fixation period before two images appeared in opposite hemifields and the fixation point disappeared. The monkey was then able to view the two images freely for the next several seconds. Each image was drawn from one of two image sets, both sampled from a series of natural image databases (see chapter 4 for more details on the images). One set consisted of images the monkey had never seen before and would never see again after a particular experimental session; the other set consisted of images that the animal had seen over one thousand times previously.

The behavior of both monkeys used in these experiments (B and S) were significantly modulated by familiarity, similarly the behavior of the monkeys described in chapter 4. However, in contrast to the neurons in LIP, the neurons in ITC had much stronger representations of image familiarity than in LIP. Numerous individual neurons in ITC signaled the familiarity of the image in its preferred hemifield prior to the animal's first saccade – and, further, a significant modulation in population average firing rates emerges prior to the first saccade in the free viewing period, much earlier than reliable modulation by familiarity in LIP. Thus, these additional experiments provide evidence that ITC is more closely involved in behavior during the PLT than LIP. While this is unsurprising given ITC's known role in the representation of image familiarity, it leaves open questions about how these undirected, familiarity-related behaviors are mediated.

Taken together, our results suggest that undirected behaviors may be mediated through an alternative pathway, potentially mediated by the frontal eye fields (FEF), due to its direct role in saccade production[24] and previously described connectivity[216] and functional interactions[217] with ITC. Our results suggest that the dorsal-ventral dichotomy may also reflect a difference in how directed and undirected tasks are mediated. In particular, rather than underlying visually-guided actions in general, the dorsal stream may be primarily involved in directed actions, while undirected actions are mediated primarily through the ventral stream and prefrontal areas, like FEF, which are known to receive inputs from both of the canonical visual streams. Further work is required to elucidate the role of ITC in undirected actions in general, as well as the role of FEF in behaviors like the PLT.

5.5 Final remarks

One of our central insights into the brain is its modular structure. This modularity is present in anatomy, and – to some degree – in function. However, much remains to be clarified about the functional significance of this modularity. Further, we lack a well-developed understanding of the forces that shape the brain to be modular. Developing a theoretical understanding of the pressures toward modularity, and coupling this new theoretical insight with targeted experiments to more precisely discover the distinct functions of anatomically distinct brain regions, is essential to progress in neuroscience. A holistic understanding of modularity will provide constraints on representations and dynamics both within and across brain regions, and thus will be extremely useful for interpreting future experimental results. The work described here has pushed on this problem from multiple directions, and we believe that the future directions described above will again move us closer toward this deeper understanding of distributed computation in neural systems.

REFERENCES

1. Carr, C. E. & Konishi, M. Axonal delay lines for time measurement in the owl's brainstem. *Proceedings of the National Academy of Sciences* **85,** 8311–8315 (1988).
2. Carr, C. & Konishi, M. A circuit for detection of interaural time differences in the brain stem of the barn owl. *Journal of Neuroscience* **10,** 3227–3246 (1990).
3. Brodmann, K. *Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues* (Barth, 1909).
4. Von Economo, C. F. & Koskinas, G. N. *Die cytoarchitektonik der hirnrinde des erwachsenen menschen* (J. Springer, 1925).
5. Felleman, D. J. & Van Essen, D. C. *Distributed hierarchical processing in the primate cerebral cortex* in *Cereb cortex* (1991).
6. Van Essen, D. C., Anderson, C. H. & Felleman, D. J. Information processing in the primate visual system: an integrated systems perspective. *Science* **255,** 419–423 (1992).
7. Van Essen, D. C. & Maunsell, J. H. Hierarchical organization and functional streams in the visual cortex. *Trends in neurosciences* **6,** 370–375 (1983).
8. Maunsell, J. H. & Van Essen, D. C. Functional properties of neurons in middle temporal visual area of the macaque monkey. I. Selectivity for stimulus direction, speed, and orientation. *Journal of neurophysiology* **49,** 1127–1147 (1983).
9. Desimone, R. & Schein, S. J. Visual properties of neurons in area V4 of the macaque: sensitivity to stimulus form. *Journal of neurophysiology* **57,** 835–868 (1987).
10. Gross, C. G., Rocha-Miranda, C. d. & Bender, D. Visual properties of neurons in inferotemporal cortex of the macaque. *Journal of neurophysiology* **35,** 96–111 (1972).
11. Barash, S., Bracewell, R. M., Fogassi, L., Gnadt, J. W. & Andersen, R. A. Saccade-related activity in the lateral intraparietal area. I. Temporal properties; comparison with area 7a. *J Neurophysiol* **66,** 1095–1108. ISSN: 00223077. `http://www.ncbi.nlm.nih.gov/pubmed/1753276` (1991).
12. Cowey, A. & Gross, C. Effects of foveal prestriate and inferotemporal lesions on visual discrimination by rhesus monkeys. *Experimental brain research* **11,** 128–144 (1970).
13. Lamotte, R. H., Acun, C., *et al.* Defects in accuracy of reaching after removal of posterior parietal cortex in monkeys. *Brain research* **139,** 309–326 (1978).
14. Parkinson, J., Murray, E. & Mishkin, M. A selective mnemonic role for the hippocampus in monkeys: memory for the location of objects. *Journal of Neuroscience* **8,** 4159–4167 (1988).
15. Baxter, M. G., Parker, A., Lindner, C. C., Izquierdo, A. D. & Murray, E. A. Control of response selection by reinforcer value requires interaction of amygdala and orbital prefrontal cortex. *Journal of Neuroscience* **20,** 4311–4319 (2000).
16. Izquierdo, A., Suda, R. K. & Murray, E. A. Bilateral orbital prefrontal cortex lesions in rhesus monkeys disrupt choices guided by both reward value and reward contingency. *Journal of Neuroscience* **24,** 7540–7548 (2004).
17. Held, R., Ingle, D., Schneider, G. & Trevarthen, C. Locating and identifying: Two modes of visual processing. *Psychologische Forschung* **31,** 42–43 (1967).
18. Ungerleider, L. G. & Mishkin, M. *Two cortical visual systems* 1982.

19. Mishkin, M., Ungerleider, L. G. & Macko, K. A. Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences* **6,** 414–417 (1983).
20. Goodale, M. A. & Westwood, D. A. An evolving view of duplex vision: Separate but interacting cortical pathways for perception and action. *Current Opinion in Neurobiology* **14,** 203–211 (2004).
21. Desimone, R., Albright, T. D., Gross, C. G. & Bruce, C. Stimulus-selective properties of inferior temporal neurons in the macaque. *Journal of Neuroscience* **4,** 2051–2062 (1984).
22. Ito, M., Tamura, H., Fujita, I. & Tanaka, K. Size and position invariance of neuronal responses in monkey inferotemporal cortex. *Journal of neurophysiology* **73,** 218–226. ISSN: 0022-3077 (1995).
23. Booth, M. C. A. & Rolls, E. T. View-invariant representations of familiar objects by neurons in the inferior temporal visual cortex. *Cerebral Cortex* **8,** 510–523. ISSN: 10473211 (1998).
24. Schiller, P., Tehovnik, E. & Larry R. Squire. Cortical Control of Eye Movements, 175–181. `http://www.sciencedirect.com.fetch.mhsl.uab.edu/science/article/B98GH-4TVBCX5-YH/2/35abdf0598d12ed33fe5e7e5f4040e3d%7B%5C%7D5Cnhttp://www.sciencedirect.com.fetch.mhsl.uab.edu/science?%7B%5C_%7Dob=ArticleURL%7B%5C&%7D%7B%5C_%7Dudi=B98GH-4TVBCX5-YH%7B%5C&%7D%7B%5C_%7Duser=446477%7B%5C&%7D%7B%5C_%7DcoverDate=03/25/2009%7B%5C&%7D%7B%5C_%7Dalid=12` (2009).
25. Andersen, R. A., Snyder, L. H., Bradley, D. C. & Xing, J. Multimodal representation of space in the posterior parietal cortex and its use in planning movements. *Annual review of neuroscience* **20,** 303–330 (1997).
26. Ipata, A. E., Gee, A. L., Goldberg, M. E. & Bisley, J. W. Activity in the Lateral Intraparietal Area Predicts the Goal and Latency of Saccades in a Free-Viewing Visual Search Task. *The Journal of Neuroscience* **26,** 3656–3661. ISSN: 0270-6474, 1529-2401 (2006).
27. Ipata, A. E., Gee, A. L., Bisley, J. W. & Goldberg, M. E. Neurons in the lateral intraparietal area create a priority map by the combination of disparate signals. *Experimental brain research* **192,** 479–488 (2009).
28. Lehky, S. R. & Sereno, A. B. Comparison of shape encoding in primate dorsal and ventral visual pathways. *Journal of neurophysiology* **97,** 307–19. ISSN: 0022-3077 (2007).
29. Van Dromme, I. C., Premereur, E., Verhoef, B.-E., Vanduffel, W. & Janssen, P. Posterior parietal cortex drives inferotemporal activations during three-dimensional object vision. *PLoS biology* **14,** e1002445 (2016).
30. Janssen, P., Verhoef, B.-E. & Premereur, E. Functional interactions between the macaque dorsal and ventral visual pathways during three-dimensional object vision. *Cortex* **98,** 218–227 (2018).
31. Sereno, A. B. & Maunsell, J. H. Shape selectivity in primate lateral intraparietal cortex. *Nature* **395,** 500–503. ISSN: 0028-0836 (1998).
32. Hong, Y. K., Lacefield, C. O., Rodgers, C. C. & Bruno, R. M. Sensation, movement and learning in the absence of barrel cortex. *Nature* **561,** 542–546 (2018).

33. Li, N., Daie, K., Svoboda, K. & Druckmann, S. Robust neuronal dynamics in premotor cortex during motor planning. *Nature* **532,** 459–64. ISSN: 1476-4687. http://dx.doi.org/10.1038/nature17643%7B%5C%%7D5Cnhttp://www.ncbi.nlm.nih.gov/pubmed/27074502 (2016).
34. Zhou, Y. & Freedman, D. J. Posterior parietal cortex plays a causal role in perceptual and categorical decisions. *Science* **365,** 180–185 (2019).
35. Chaudhuri, R., Knoblauch, K., Gariel, M.-A., Kennedy, H. & Wang, X.-J. A large-scale circuit mechanism for hierarchical dynamical processing in the primate cortex. *Neuron* **88,** 419–431 (2015).
36. Guo, Z. V. *et al.* Maintenance of persistent activity in a frontal thalamocortical loop. *Nature* **545,** 181–186 (2017).
37. Rust, N. C. & DiCarlo, J. J. Balanced increases in selectivity and tolerance produce constant sparseness along the ventral visual stream. *J Neurosci* **32,** 10170–10182. ISSN: 1529-2401. arXiv: arXiv:1011.1669v3. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve%7B%5C&%7Ddb=PubMed%7B%5C&%7Ddopt=Citation%7B%5C&%7Dlist%7B%5C_%7Duids=22836252 (2012).
38. DiCarlo, J. J., Zoccolan, D. & Rust, N. C. How does the brain solve visual object recognition? *Neuron* **73,** 415–434 (2012).
39. Yamins, D. L. *et al.* Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences* **111,** 8619–8624 (2014).
40. Yamins, D. L. & DiCarlo, J. J. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience* **19,** 356–365 (2016).
41. Pagan, M., Urban, L. S., Wohl, M. P. & Rust, N. C. Signals in inferotemporal and perirhinal cortex suggest an untangling of visual target information. *Nature neuroscience* **16,** 1132–1139 (2013).
42. Chung, S., Lee, D. D. & Sompolinsky, H. Classification and geometry of general perceptual manifolds. *Physical Review X* **8,** 031003 (2018).
43. Cohen, U., Chung, S., Lee, D. D. & Sompolinsky, H. Separability and geometry of object manifolds in deep neural networks. *Nature communications* **11,** 1–13 (2020).
44. Gallego, J. A. *et al.* Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature communications* **9,** 1–13 (2018).
45. Vyas, S., Golub, M. D., Sussillo, D. & Shenoy, K. V. Computation Through Neural Population Dynamics. *Annual Review of Neuroscience* **43,** 249–275 (2020).
46. Barlow, H. B. Possible principles underlying the transformations of sensory messages. *Sensory Communication,* 217–234 (1961).
47. Barlow, H. B. Redundancy reduction revisited. *Network* **12,** 241–253 (2001).
48. Laughlin, S. A simple coding procedure enhances a neuron's information capacity. *Zeitschrift für Naturforschung c* **36,** 910–912 (1981).
49. Dan, Y., Atick, J. J. & Reid, R. C. Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. *Journal of Neuroscience* **16,** 3351–3362 (1996).

50. Fairhall, A. L., Lewen, G. D., Bialek, W. & van Steveninck, R. R. d. R. Efficiency and ambiguity in an adaptive neural code. *Nature* **412,** 787–792 (2001).
51. Olshausen, B. A. & Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381,** 607–609 (1996).
52. Olshausen, B. A. & Field, D. J. Sparse coding of sensory inputs. *Current Opinion in Neurobiology* **14,** 481–487 (2004).
53. Faisal, A. A., Selen, L. P. & Wolpert, D. M. Noise in the nervous system. *Nature reviews neuroscience* **9,** 292 (2008).
54. Renart, A. & Machens, C. K. Variability in neural activity and behavior. *Current opinion in neurobiology* **25,** 211–220 (2014).
55. Stringer, C. *et al.* Spontaneous behaviors drive multidimensional, brainwide activity. *Science* **364** (2019).
56. Musall, S., Kaufman, M. T., Juavinett, A. L., Gluf, S. & Churchland, A. K. Single-trial neural dynamics are dominated by richly varied movements. *bioRxiv,* 308288 (2019).
57. Sreenivasan, S. & Fiete, I. Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nature Neuroscience* **14,** 1330–1337. `http://dx.doi.org/10.1038/nn.2901` (2011).
58. Stringer, C., Pachitariu, M., Steinmetz, N., Carandini, M. & Harris, K. D. High-dimensional geometry of population responses in visual cortex. *Nature,* 1 (2019).
59. Johnston, W. J., Palmer, S. E. & Freedman, D. J. Nonlinear mixed selectivity supports reliable neural computation. *PLoS computational biology* **16,** e1007544 (2020).
60. MacKay, D. J. *Information theory, inference and learning algorithms* (Cambridge University Press, 2003).
61. Cover, T. M. & Thomas, J. A. *Elements of information theory* (John Wiley & Sons, 2012).
62. Rigotti, M. *et al.* The importance of mixed selectivity in complex cognitive tasks. *Nature* **497,** 1–6. `http://dx.doi.org/10.1038/nature12160%7B%5C%%7D5Cnpapers3://publication/doi/10.1038/nature12160` (2013).
63. Fusi, S., Miller, E. K. & Rigotti, M. Why neurons mix: High dimensionality for higher cognition. *Current Opinion in Neurobiology* **37,** 66–74. `http://dx.doi.org/10.1016/j.conb.2016.01.010` (2016).
64. Hubel, D. H. & Wiesel, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology* **195,** 215–243 (1968).
65. Finkelstein, A., Ulanovsky, N., Tsodyks, M. & Aljadeff, J. Optimal dynamic coding by mixed-dimensionality neurons in the head-direction system of bats. *Nature communications* **9,** 3590 (2018).
66. Eichler, K. *et al.* The complete connectome of a learning and memory centre in an insect brain. *Nature* **548,** 175 (2017).
67. Sosulski, D. L., Bloom, M. L., Cutforth, T., Axel, R. & Datta, S. R. Distinct representations of olfactory information in different cortical centres. *Nature* **472,** 213 (2011).

68. Walker, K. M., Bizley, J. K., King, A. J. & Schnupp, J. W. Multiplexed and robust representations of sound features in auditory cortex. *Journal of Neuroscience* **31,** 14565–14576 (2011).
69. Petersen, R. S. *et al.* Diverse and temporally precise kinetic feature selectivity in the VPm thalamic nucleus. *Neuron* **60,** 890–903 (2008).
70. Churchland, M. M. & Shenoy, K. V. Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *Journal of neurophysiology* **97,** 4235–4257 (2007).
71. Hatsopoulos, N. G., Xu, Q. & Amit, Y. Encoding of movement fragments in the motor cortex. *Journal of Neuroscience* **27,** 5105–5114 (2007).
72. Sergio, L. E. & Kalaska, J. F. Changes in the temporal pattern of primary motor cortex activity in a directional isometric force versus limb movement task. *Journal of neurophysiology* **80,** 1577–1583 (1998).
73. Curto, C., Itskov, V., Morrison, K., Roth, Z. & Walker, J. L. Combinatorial neural codes from a mathematical coding theory perspective. *Neural computation* **25,** 1891–1925 (2013).
74. Barak, O., Rigotti, M. & Fusi, S. The sparseness of mixed selectivity neurons controls the generalization–discrimination trade-off. *Journal of Neuroscience* **33,** 3844–3856 (2013).
75. Bernardi, S. *et al.* The geometry of abstraction in hippocampus and pre-frontal cortex. *bioRxiv,* 408633 (2018).
76. Rubin, D. B., Van Hooser, S. D. & Miller, K. D. The Stabilized Supralinear Network: A Unifying Circuit Motif Underlying Multi-Input Integration in Sensory Cortex. *Neuron* **85,** 402–417. ISSN: 08966273 (2015).
77. Wang, X.-J. Decision making in recurrent neuronal circuits. *Neuron* **60,** 215–234 (2008).
78. Compte, A., Brunel, N., Goldman-Rakic, P. S. & Wang, X.-J. Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral cortex* **10,** 910–923 (2000).
79. Brunel, N. Dynamics and plasticity of stimulus-selective persistent activity in cortical network models. *Cerebral Cortex* **13,** 1151–1161 (2003).
80. Joglekar, M. R., Mejias, J. F., Yang, G. R. & Wang, X.-J. Inter-areal balanced amplification enhances signal propagation in a large-scale circuit model of the primate cortex. *Neuron* **98,** 222–234 (2018).
81. Gu, Y., Angelaki, D. E. & DeAngelis, G. C. Neural correlates of multisensory cue integration in macaque MSTd. *Nature neuroscience* **11,** 1201–1210 (2008).
82. Fetsch, C. R., DeAngelis, G. C. & Angelaki, D. E. Visual–vestibular cue integration for heading perception: applications of optimal cue integration theory. *European Journal of Neuroscience* **31,** 1721–1729 (2010).
83. Deneve, S., Latham, P. E. & Pouget, A. Efficient computation and cue integration with noisy population codes. *Nature Neuroscience* **4,** 826–831. ISSN: 1097-6256. `http://www.ncbi.nlm.nih.gov/pubmed/11477429` (2001).

84. Avillac, M., Deneve, S., Olivier, E., Pouget, A. & Duhamel, J.-R. Reference frames for representing visual and tactile locations in parietal cortex. *Nature neuroscience* **8,** 941–949 (2005).
85. Olivier, E., Pouget, A., Avillac, M. & Dene, S. Reference frames for representing visual and tactile locations in parietal cortex. **8,** 941–949 (2005).
86. Dokka, K., Park, H., Jansen, M., DeAngelis, G. C. & Angelaki, D. E. Causal inference accounts for heading perception in the presence of object motion. *Proceedings of the National Academy of Sciences* **116,** 9060–9065 (2019).
87. Dokka, K., DeAngelis, G. C. & Angelaki, D. E. Multisensory integration of visual and vestibular signals improves heading discrimination in the presence of a moving object. *Journal of Neuroscience* **35,** 13599–13607 (2015).
88. Körding, K. P. *et al.* Causal inference in multisensory perception. *PLoS one* **2,** e943 (2007).
89. Pouget, A., Deneve, S. & Duhamel, J.-R. A computational perspective on the neural basis of multisensory spatial representations. *Nature Reviews Neuroscience* **3,** 741–747 (2002).
90. Treisman, A. M. & Gelade, G. A feature-integration theory of attention. *Cognitive psychology* **12,** 97–136 (1980).
91. Quinlan, P. T. Visual feature integration theory: Past, present, and future. *Psychological bulletin* **129,** 643 (2003).
92. Schneegans, S. & Bays, P. M. Neural architecture for feature binding in visual working memory. *Journal of Neuroscience* **37,** 3913–3925 (2017).
93. Krakauer, J. W., Ghazanfar, A. A., Gomez-Marin, A., Maciver, M. A. & Poeppel, D. Neuron Perspective Neuroscience Needs Behavior: Correcting a Reductionist Bias. *Neuron* **93,** 480–490. ISSN: 08966273. `http://dx.doi.org/10.1016/j.neuron.2016.12.041` (2017).
94. Rust, N. C. & Movshon, J. A. In praise of artifice. *Nature neuroscience* **8,** 1647–1650. ISSN: 1097-6256 (2005).
95. Hessler, N. A. & Doupe, A. J. Social context modulates singing-related neural activity in the songbird forebrain. *Nature neuroscience* **2,** 209–211 (1999).
96. Kao, M. H., Doupe, A. J. & Brainard, M. S. Contributions of an avian basal ganglia–forebrain circuit to real-time modulation of song. *Nature* **433,** 638–643 (2005).
97. Niell, C. M. & Stryker, M. P. Modulation of visual responses by behavioral state in mouse visual cortex. *Neuron* **65,** 472–479 (2010).
98. Dadarlat, M. C. & Stryker, M. P. Locomotion enhances neural encoding of visual stimuli in mouse V1. *Journal of Neuroscience* **37,** 3764–3775 (2017).
99. Buschman, T. J. & Miller, E. K. Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *Science (New York, N.Y.)* **315,** 1860–1862. ISSN: 0036-8075 (2007).
100. Hyvarinen, A. & Oja, E. Independent component analysis: Algorithms and applications. *Neural Networks* **13,** 411–430 (2000).
101. Gardner-Medwin, A. & Barlow, H. B. The limits of counting accuracy in distributed neural representations. *Neural Computation* **13,** 477–504 (2001).

102. Vinje, W. E. & Gallant, J. L. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science* **287,** 1273–1276 (2000).
103. Lewicki, M. S. Efficient coding of natural sounds. *Nature Neuroscience* **5** (2002).
104. Smith, E. C. & Lewicki, M. S. Efficient auditory coding. *Nature* **439,** 978–982 (2006).
105. Perez-Orive, J. *et al.* Oscillations and Sparsening of Odor Representations in the Mushroom Body. *Science* **297,** 359–365 (2002).
106. Koyluoglu, O. O., Pertzov, Y., Manohar, S., Husain, M. & Fiete, I. R. Fundamental bound on the persistence and capacity of short-term memory stored as graded persistent activity. *eLife* **6,** e22225 (2017).
107. Linsker, R. How to generate ordered maps by maximizing the mutual information between input and output signals. *Neural computation* **1,** 402–411 (1989).
108. Linsker, R. Perceptual neural organization: Some approaches based on network models and information theory. *Annual review of Neuroscience* **13,** 257–281 (1990).
109. Haft, M. & Van Hemmen, J. Theory and implementation of infomax filters for the retina. *Network: Computation in Neural Systems* **9,** 39–71 (1998).
110. Okajima, K. Two-dimensional Gabor-type receptive field as derived by mutual information maximization. *Neural Networks* **11,** 441–447 (1998).
111. Zhang, K. & Sejnowski, T. J. Neuronal tuning: To sharpen or broaden? *Neural computation* **11,** 75–84. arXiv: `arXiv:1011.1669v3`. `http://www.ncbi.nlm.nih.gov/pubmed/9950722` (1999).
112. Eurich, C. W. & Wilke, S. D. Multidimensional Encoding Strategy of Spiking Neurons. *Neural Computation* **1529,** 1519–1529 (2000).
113. Wang, Z., Wei, X.-X., Stocker, A. A. & Lee, D. D. *Efficient neural codes under metabolic constraints* in *Advances in Neural Information Processing Systems* (2016), 4619–4627.
114. Brunel, N. & Nadal, J.-P. P. Mutual information, Fisher information, and population coding. *Neural computation* **10,** 1731–1757. arXiv: `arXiv:1011.1669v3`. `http://www.ncbi.nlm.nih.gov/pubmed/9744895` (1998).
115. Wei, X.-X. & Stocker, A. A. Mutual information, Fisher information, and efficient coding. *Neural computation* **28,** 305–326 (2016).
116. Park, I. M. & Pillow, J. W. Bayesian efficient coding. *bioRxiv,* 178418 (2017).
117. Kulldorff, G. On the conditions for consistency and asymptotic efficiency of maximum likelihood estimates. *Scandinavian Actuarial Journal* **1957,** 129–144 (1957).
118. Bethge, M., Rotermund, D. & Pawelzik, K. Optimal short-term population coding: when Fisher information fails. *Neural computation* **14,** 2317–2351 (2002).
119. Resulaj, A., Ruediger, S., Olsen, S. R. & Scanziani, M. First spikes in visual cortex enable perceptual discrimination. *eLife* **7,** e34044 (2018).
120. Litwin-Kumar, A., Harris, K. D., Axel, R., Sompolinsky, H. & Abbott, L. Optimal Degrees of Synaptic Connectivity. *Neuron* **0,** 1153–1164.e7. `http://linkinghub.elsevier.com/retrieve/pii/S0896627317300545` (2017).
121. Alemi, A. & Abbara, A. Exponential Capacity in an Autoencoder Neural Network with a Hidden Layer. *arXiv.* arXiv: `arXiv:1705.07441v1` (2017).

122. Tootoonian, S. & Lengyel, M. *A dual algorithm for olfactory computation in the locust brain* in *Advances in neural information processing systems* (2014), 2276–2284.
123. Zwicker, D., Murugan, A. & Brenner, M. P. Receptor arrays optimized for natural odor statistics. *Proceedings of the National Academy of Sciences* **113,** 5570–5575 (2016).
124. Zhang, Y. & Sharpee, T. O. A robust feedforward model of the olfactory system. *PLoS computational biology* **12,** e1004850 (2016).
125. Shlens, J. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100* (2014).
126. Moreno-Bote, R. *et al.* Information-limiting correlations. *Nature neuroscience* **17,** 1410 (2014).
127. Laughlin, S. B. Energy as a constraint on the coding and processing of sensory information. *Current opinion in neurobiology* **11,** 475–480 (2001).
128. Schüz, A. & Palm, G. Density of neurons and synapses in the cerebral cortex of the mouse. *Journal of Comparative Neurology* **286,** 442–455 (1989).
129. Levy, W. B. & Baxter, R. A. Energy efficient neural codes. *Neural computation* **8,** 531–543 (1996).
130. Laughlin, S. B. & Sejnowski, T. J. Communication in Neuronal Networks. *Science* **301,** 1870–1875 (2003).
131. Olshausen, B. A. & Field, D. J. in *Problems in Systems Neuroscience* (eds van Hemmen, J. L. & Sejnowski, T.) 182–211 (Oxford University Press, 2006).
132. McClelland, J. L., Rumelhart, D. E., Group, P. R., *et al. Parallel distributed processing* (MIT press Cambridge, MA: 1987).
133. Eurich, C. W. & Schwegler, H. Coarse coding: calculation of the resolution achieved by a population of large receptive field neurons. *Biological cybernetics* **76,** 357–363 (1997).
134. Gross, C. G., Bender, D. B. & Rocha-Miranda, C. E. Visual receptive fields of neurons in inferotemporal cortex of the monkey. *Science* **166,** 1303–6. ISSN: 0036-8075. `http://www.ncbi.nlm.nih.gov/pubmed/4982685` (1969).
135. Rishel, C. A., Huang, G. & Freedman, D. J. Independent category and spatial encoding in parietal cortex. *Neuron* **77,** 969–979. `http://dx.doi.org/10.1016/j.neuron.2013.01.007` (2013).
136. Spanne, A. & Jörntell, H. Questioning the role of sparse coding in the brain. *Trends in neurosciences* **38,** 417–427 (2015).
137. Brown, W. M. & Bäcker, A. Optimal neuronal tuning for finite stimulus spaces. *Neural computation* **18,** 1511–1526 (2006).
138. Lindsay, G. W., Rigotti, M., Warden, M. R., Miller, E. K. & Fusi, S. Hebbian learning in a random network captures selectivity properties of the prefrontal cortex. *Journal of Neuroscience* **37,** 11021–11036 (2017).
139. Babadi, B. & Sompolinsky, H. Sparseness and expansion in sensory representations. *Neuron* **83,** 1213–1226 (2014).
140. Finkelstein, A. *et al.* Three-dimensional head-direction coding in the bat brain. *Nature* **517,** 159 (2015).

141. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503,** 78–84. arXiv: 15334406. http://dx.doi.org/10.1038/nature12742 (2013).
142. Knill, D. C. & Pouget, A. The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends in Neurosciences* **27,** 712–719 (2004).
143. Zemel, R. S., Dayan, P. & Pouget, A. Probabilistic interpretation of population codes. *Neural computation* **10,** 403–430 (1998).
144. Saunders, J. A. & Knill, D. C. Perception of 3D surface orientation from skew symmetry. *Vision research* **41,** 3163–3183 (2001).
145. Jones, E., Oliphant, T., Peterson, P., *et al. SciPy: Open source scientific tools for Python* 2001–. http://www.scipy.org/.
146. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011).
147. Atick, J. J. & Redlich, A. N. Towards a theory of early visual processing. *Neural Computation* **2,** 308–320 (1990).
148. Shannon, C. E. Probability of error for optimal codes in a Gaussian channel. *Bell System Technical Journal* **38,** 611–656 (1959).
149. Balasubramanian, V. & Berry, M. J. A test of metabolically efficient coding in the retina. *Network: Computation in Neural Systems* **13,** 531–552 (2002).
150. Churchland, M. M. *et al.* Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature Neuroscience* **13,** 369–378. http://dx.doi.org/10.1038/nn.2501 (2010).
151. Arimoto, S. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory* **18,** 14–20 (1972).
152. Blahut, R. Computation of channel capacity and rate-distortion functions. *IEEE transactions on Information Theory* **18,** 460–473 (1972).
153. Köhler, W. Gestalt psychology. *Psychologische Forschung* **31,** XVIII–XXX (1967).
154. Wagemans, J. *et al.* A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization. *Psychological bulletin* **138,** 1172 (2012).
155. Zhou, H., Friedman, H. S. & Von Der Heydt, R. Coding of border ownership in monkey visual cortex. *Journal of Neuroscience* **20,** 6594–6611 (2000).
156. Lamme, V. A., Rodriguez-Rodriguez, V. & Spekreijse, H. Separate processing dynamics for texture elements, boundaries and surfaces in primary visual cortex of the macaque monkey. *Cerebral cortex* **9,** 406–413 (1999).
157. Treisman, A., Schmidt, H., *et al.* Illusory conjunctions in the perception of objects. *1982* **14,** 107–141 (1982).
158. Ivry, R. B. & Prinzmetal, W. Effect of feature similarity on illusory conjunctions. *Perception & psychophysics* **49,** 105–116 (1991).
159. Makous, J. C. & Middlebrooks, J. C. Two-dimensional sound localization by human listeners. *The journal of the Acoustical Society of America* **87,** 2188–2200 (1990).
160. Zhang, W. & Luck, S. J. Discrete fixed-resolution representations in visual working memory. *Nature* **453,** 233–235 (2008).

161. Vehtari, A., Gelman, A. & Gabry, J. Efficient implementation of leave-one-out cross-validation and WAIC for evaluating fitted Bayesian models. *arXiv preprint arXiv:1507.04544* (2015).
162. Vehtari, A., Gelman, A. & Gabry, J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and computing* **27,** 1413–1432 (2017).
163. Orhan, A. E. & Ma, W. J. Neural population coding of multiple stimuli. *Journal of Neuroscience* **35,** 3825–3841 (2015).
164. Bays, P. M., Catalao, R. F. & Husain, M. The precision of visual working memory is set by allocation of a shared resource. *Journal of vision* **9,** 7–7 (2009).
165. Van den Berg, R., Awh, E. & Ma, W. J. Factorial comparison of working memory models. *Psychological review* **121,** 124 (2014).
166. Carpenter, B. *et al.* Stan: A probabilistic programming language. *Journal of statistical software* **76** (2017).
167. Kreft, I. G. & De Leeuw, J. *Introducing multilevel modeling* (Sage, 1998).
168. Gelman, A. Multilevel (hierarchical) modeling: what it can and cannot do. *Technometrics* **48,** 432–435 (2006).
169. Desimone, R. & Duncan, J. Neural mechanisms of selective visual attention. *Annual review of neuroscience* **18,** 193–222 (1995).
170. Ungerleider, S. K. & G, L. Mechanisms of visual attention in the human cortex. *Annual review of neuroscience* **23,** 315–341 (2000).
171. Teller, D. Y. The forced-choice preferential looking procedure: A psychophysical technique for use with human infants. *Infant Behavior and Development* **2,** 135–153 (1979).
172. Jutras, M. J. & Buffalo, E. A. Recognition memory signals in the macaque hippocampus. *Proceedings of the National Academy of Sciences* **107,** 401–406 (2010).
173. Manns, J. R., Stark, C. E. & Squire, L. R. The visual paired-comparison task as a measure of declarative memory. *Proceedings of the National Academy of Sciences* **97,** 12375–12379 (2000).
174. Pascalis, O. & Bachevalier, J. Face recognition in primates: a cross-species study. *Behavioural processes* **43,** 87–96 (1998).
175. Wang, Q., Cavanagh, P. & Green, M. Familiarity and pop-out in visual search. *Perception & psychophysics* **56,** 495–500 (1994).
176. Shen, J. & Reingold, E. M. Visual search asymmetry: The influence of stimulus familiarity and low-level features. *Perception & Psychophysics* **63,** 464–475 (2001).
177. Malinowski, P. & Hübner, R. The effect of familiarity on visual-search performance: Evidence for learned basic features. *Perception & Psychophysics* **63,** 458–463 (2001).
178. Bichot, N. P., Schall, J. D. & Thompson, K. G. *Visual feature selectivity in frontal eye fields induced by experience in mature macaques.* 1996.
179. Horowitz, T. S. & Wolfe, J. M. Visual search has no memory. *Nature* **394,** 575–577 (1998).
180. Wardak, C., Olivier, E. & Duhamel, J.-R. Saccadic target selection deficits after lateral intraparietal area inactivation in monkeys. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **22,** 9877–84. ISSN: 1529-2401. `http://www.ncbi.nlm.nih.gov/pubmed/12427844` (2002).

181. Beck, D. M., Rees, G., Frith, C. D. & Lavie, N. Neural correlates of change detection and change blindness. *Nature neuroscience* **4,** 645–650 (2001).
182. Cohen, M. R. & Maunsell, J. H. Attention improves performance primarily by reducing interneuronal correlations. *Nature neuroscience* **12,** 1594 (2009).
183. Bisley, J. W. & Mirpour, K. The neural instantiation of a priority map. *Current opinion in psychology* **29,** 108–112 (2019).
184. Zhou, H. & Desimone, R. Feature-based attention in the frontal eye field and area V4 during visual search. *Neuron* **70,** 1205–1217 (2011).
185. Bisley, J. W., Mirpour, K., Arcizet, F. & Ong, W. S. The role of the lateral intraparietal area in orienting attention and its implications for visual search. *European Journal of Neuroscience* **33,** 1982–1990 (2011).
186. Wardak, C., Ibos, G., Duhamel, J.-R. & Olivier, E. Contribution of the monkey frontal eye field to covert visual attention. *Journal of Neuroscience* **26,** 4228–4235 (2006).
187. Buschman, T. J. & Miller, E. K. Serial, covert shifts of attention during visual search are reflected by the frontal eye fields and correlated with population oscillations. *Neuron* **63,** 386–396 (2009).
188. Herrington, T. M. & Assad, J. A. Temporal Sequence of Attentional Modulation in the Lateral Intraparietal Area and Middle Temporal Area during Rapid Covert Shifts of Attention. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **30,** 3287–3296. ISSN: 0270-6474 (2010).
189. Freedman, D. J. & Assad, J. A. Experience-dependent representation of visual categories in parietal cortex. *Nature* **443,** 85 (2006).
190. Grunewald, A., Linden, J. F. & Andersen, R. A. Responses to auditory stimuli in macaque lateral intraparietal area I. Effects of training. *Journal of neurophysiology* **82,** 330–342 (1999).
191. Sarma, A., Masse, N. Y., Wang, X.-J. & Freedman, D. J. Task-specific versus generalized mnemonic representations in parietal and prefrontal cortices. *Nature neuroscience* **19,** 143–149. ISSN: 1546-1726 (2016).
192. Swaminathan, S. K. & Freedman, D. J. Preferential encoding of visual categories in parietal cortex compared with prefrontal cortex. *Nature neuroscience* **15,** 315–320 (2012).
193. Ibos, G. & Freedman, D. J. Dynamic integration of task-relevant visual features in posterior parietal cortex. *Neuron* **83,** 1468–1480 (2014).
194. Christopoulos, V. N., Kagan, I. & Andersen, R. A. Lateral intraparietal area (LIP) is largely effector-specific in free-choice decisions. *Scientific reports* **8,** 1–13 (2018).
195. Kobak, D. *et al.* Demixed principal component analysis of neural population data. *eLife* **5,** 1–36. ISSN: 2050084X. arXiv: 1410.6031 (2016).
196. Marcos, A. S. & Harvey, C. D. History-dependent variability in population dynamics during evidence accumulation in cortex. *Nature neuroscience* **19,** 1672–1680. ISSN: 1097-6256 (2016).
197. Fahy, F., Riches, I. & Brown, M. Neuronal activity related to visual recognition memory: long-term memory and the encoding of recency and familiarity information in the

primate anterior and medial inferior temporal and rhinal cortex. *Experimental Brain Research* **96,** 457–472 (1993).

198. Huang, G., Ramachandran, S., Lee, T. S. & Olson, C. R. Neural correlate of visual familiarity in macaque area V2. *Journal of Neuroscience* **38,** 8967–8975 (2018).
199. Foley, N. C., Jangraw, D. C., Peck, C. & Gottlieb, J. Novelty enhances visual salience independently of reward in the parietal lobe. *J Neurosci* **34,** 7947–7957. ISSN: 1529-2401. `http://www.ncbi.nlm.nih.gov/pubmed/24899716` (2014).
200. Freedman, D. J., Riesenhuber, M., Poggio, T. & Miller, E. K. Experience-dependent sharpening of visual shape selectivity in inferior temporal cortex. *Cerebral Cortex* **16,** 1631–1644 (2006).
201. Anderson, B., Mruczek, R. E. B., Kawasaki, K. & Sheinberg, D. Effects of familiarity on neural activity in monkey inferior temporal lobe. *Cerebral Cortex* **18,** 2540–2552. ISSN: 10473211. arXiv: `NIHMS150003` (2008).
202. Woloszyn, L. & Sheinberg, D. L. L. Effects of Long-Term Visual Experience on Responses of Distinct Classes of Single Units in Inferior Temporal Cortex. *Neuron* **74,** 193–205. ISSN: 08966273 (2012).
203. Chen, X. *et al.* Parietal Cortex Regulates Visual Salience and Salience-Driven Behavior. *Neuron* (2020).
204. Swaminathan, S. K., Masse, N. Y. & Freedman, D. J. A comparison of lateral and medial intraparietal areas during a visual categorization task. *Journal of Neuroscience* **33,** 13157–13170 (2013).
205. Asaad, W. F., Santhanam, N., McClellan, S. & Freedman, D. J. High-performance execution of psychophysical tasks with complex visual stimuli in MATLAB. *Journal of neurophysiology* **109,** 249–260 (2013).
206. Hwang, J., Mitz, A. R. & Murray, E. A. NIMH MonkeyLogic: Behavioral control and data acquisition in MATLAB. *Journal of neuroscience methods* **323,** 13–21 (2019).
207. Gnadt, J. W. & Andersen, R. A. Memory related motor planning activity in posterior parietal cortex of macaque. *Experimental brain research* **70,** 216–220 (1988).
208. Roitman, J. D. & Shadlen, M. N. Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. *Journal of neuroscience* **22,** 9475–9489 (2002).
209. Pachitariu, M., Steinmetz, N., Kadir, S., Carandini, M. & Harris, K. Fast and accurate spike sorting of high-channel count probes with KiloSort, 1–9 (2016).
210. Borji, A., Cheng, M.-M., Jiang, H. & Li, J. Salient object detection: A benchmark. *IEEE transactions on image processing* **24,** 5706–5722 (2015).
211. Zhang, J. & Sclaroff, S. *Saliency detection: A boolean map approach* in *Proceedings of the IEEE international conference on computer vision* (2013), 153–160.
212. Mainen, Z. & Sejnowski, T. Reliability of spike timing in neocortical neurons. *Science* **268,** 1503–1506. ISSN: 0036-8075. eprint: `https://science.sciencemag.org/content/268/5216/1503.full.pdf`. `https://science.sciencemag.org/content/268/5216/1503` (1995).
213. Sober, S. J., Sponberg, S., Nemenman, I. & Ting, L. H. Millisecond Spike Timing Codes for Motor Control. *Trends in Neurosciences* **41.** Special Issue: Time in the Brain,

644–648. ISSN: 0166-2236. http://www.sciencedirect.com/science/article/pii/S0166223618302285 (2018).

214. Joseph, A. & Barron, A. R. Least squares superposition codes of moderate dictionary size are reliable at rates up to capacity. *IEEE Transactions on Information Theory* **58,** 2541–2557 (2012).
215. Matthey, L., Bays, P. M. & Dayan, P. A probabilistic palimpsest model of visual short-term memory. *PLoS computational biology* **11,** e1004003 (2015).
216. Schall, J. D., Morel, A., King, D. J. & Bullier, J. Topography of visual cortex connections with frontal eye field in macaque: convergence and segregation of processing streams. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **15,** 4464–4487. ISSN: 0270-6474 (1995).
217. Monosov, I. E., Sheinberg, D. L. & Thompson, K. G. The effects of prefrontal cortex inactivation on object responses of single neurons in the inferotemporal cortex during visual search. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **31,** 15956–61. ISSN: 1529-2401. http://www.jneurosci.org/content/31/44/15956.abstract?etoc (2011).