The Inevitability of Probability: Probabilistic Inference in Generic Neural Networks Trained with Non-Probabilistic Feedback

A. Emin Orhan[†] eorhan@cns.nyu.edu

Wei Ji Ma[†,‡] weijima@nyu.edu

[†]Center for Neural Science and [‡]Department of Psychology
New York University, New York, NY 10003

January 14, 2016

Abstract

Humans and other animals have been shown to perform near-optimal probabilistic inference in a wide range of psychophysical tasks, from causal inference to cue combination to visual search. On the face of it, this is surprising because optimal probabilistic inference in each case is associated with highly non-trivial behavioral strategies. Yet, typically subjects receive little to no feedback during most of these tasks and the received feedback is not explicitly probabilistic in nature. How can subjects learn such non-trivial behavioral strategies from scarce non-probabilistic feedback? We show that generic feed-forward and recurrent neural networks trained with a relatively small number of non-probabilistic examples using simple error-based learning rules can perform near-optimal probabilistic inference in standard psychophysical tasks. The hidden layers of the trained networks develop a novel sparsity-based probabilistic population code. In all tasks, performance asymptotes at very small network sizes, usually on the order of tens of hidden units, due to the low computational complexity of the typical psychophysical tasks. For the same reason, in many cases, the trained networks also display remarkable generalization to stimulus conditions not seen during training. We further show that in a probabilistic binary categorization task involving arbitrary categories where both human and monkey subjects have been shown to perform probabilistic inference, a monkey subject's performance (but not human subjects' performance) is consistent with an error-based learning rule. Our results suggest that near-optimal probabilistic inference in standard psychophysical tasks emerges naturally and robustly in generic neural networks trained with error-based learning rules, even when neither the training objective nor the training examples are explicitly probabilistic, and that these types of networks can be used as simple plausible neural models of probabilistic inference.

Introduction

Humans and other animals have been shown to behave in accordance with the tenets of probability theory in dealing with uncertain sensory information in a wide variety of psychophysical tasks. At the behavioral level, this often implies non-trivial response strategies. For instance, in cue combination, accurate probabilistic inference requires the observer to determine the reliabilities of individual sensory cues and adjust their weights accordingly on a trial by trial basis. How can such computations be implemented in neural circuits? A prominent framework addressing this question is the probabilistic population coding (PPC) framework, according to which the population activity on a single trial encodes a probability distribution rather than a single estimate and computations with probability distributions can be carried out by suitable operations on the corresponding neural responses [23], [40]. In some cases, these operations can be as simple as a linear summation. Ma et al. [23] showed that if neural variability belongs to a particular class of probability distributions, computation of the posterior distribution in cue combination tasks can be done with a linear combination of the input responses. Moreover, in this scheme, the form of neural variability is preserved between the input and the output, leading to an elegantly modular code. This framework has since been extended to the neural implementation of more complex tasks [3], [4], [24], [25], [32]. Multiplication and division of neural responses have been argued to be necessary for optimal inference in these more complex tasks.

Upon closer look, however, this framework has several shortcomings. First, the networks in these studies were either fully manually designed, or partially manually designed and partially trained with large amounts of probabilistic data to optimize explicitly probabilistic objectives, e.g. minimization of Kullback-Leibler (KL) divergence. Therefore, this literature does not address the important question of learning: how can probabilistic inference be learned from a realistic amount and type of data with minimal manual design of the networks? Second, although there are some commonalities between the neural operations required to implement probabilistic inference in different tasks, these operations generally differ from task to task. For instance, it has been argued that some form of divisive normalization of neural responses is necessary in tasks that involve marginalization [4]. However, the specific form of divisive normalization individual neurons have to perform differs substantially from task to task. Therefore, it is unclear if probabilistic inference can be implemented in *generic* neural networks that consists of neurons performing the same type of neurally plausible operation. Third, in these studies, the number of neurons required to implement a probabilistic task scales quite unfavorably with the number of input neurons. Therefore, the question of whether these tasks can be implemented more efficiently, using fewer neurons, remains open. In this paper, we address these issues.

Using a simple error-based learning rule, we trained generic feed-forward and recurrent neural networks on eight probabilistic psychophysical tasks that span a range of ecologically important computations. The networks learned to perform near-optimal probabilistic inference from a relatively small number of training trials and were able to generalize beyond the particular stimulus conditions used in training. The trained networks developed a novel sparsity-based probabilistic population code in their hidden layers where the sparsity of the hidden layer activity was significantly correlated with sensory reliability. The strength of this relationship displayed heterogeneity from task to task in a way that can be explained with a simplified model. In all tasks, the performance of the trained networks asymptoted at very small network sizes, usually on the order of only tens of hidden units, due to the low-dimensional nature of the solutions. Thus, these generic networks perform probabilistic inference much more efficiently than the earlier PPC approaches. In hindsight, the crucial problem with those approaches is that they all postulated a too literal, one-to-one correspondence between the required neural computations at the population level and the computations individual neurons have to perform. By contrast, we show here that population-level computations required for optimal probabilistic inference can be implemented much more efficiently in generic neural networks.

We further show that in a probabilistic binary categorization task involving arbitrary categories where both human and monkey subjects have been shown to perform probabilistic inference [32], a monkey subject's performance (but not human subjects' performance) is consistent with an error-based learning rule. Our results suggest that near-optimal probabilistic inference in standard psychophysical tasks emerges naturally and robustly in generic neural networks trained with error-based learning rules and that these types of networks can be used as simple plausible neural models of probabilistic inference in non-human observers.

Results

Tasks. We trained generic feed-forward or recurrent neural networks on eight probabilistic psychophysical tasks that are commonly studied in the experimental and computational literature. The tasks were: (a) linear cue combination [2], [9], [18], (b) modular cue combination, (c) coordinate transformations [4], [41], (d) Kalman filtering [20], [27], [38], (e) binary categorization [32], (f) stimulus demixing [4], (g) causal inference [19], [25], (h) visual search with heterogeneous distractors [24] (see *Methods* for task details).

Networks. The networks all received noisy sensory information about the stimulus or the stimuli in the form of a neural population with Poisson variability. The hidden units of the networks were modeled as rectified linear units (ReLUs). ReLUs are commonly used in neural networks due to their demonstrated advantage over alternative non-linearities in gradient-based learning schemes [11]. Linear (sigmoidal) output units were used in tasks with continuous (categorical) outputs. Schematic diagrams of all the networks used for the tasks considered in this paper are shown in Figure 1. Networks were trained to minimize mean squared error or cross-entropy in tasks with continuous or categorical outputs, respectively. Importantly, the networks were provided only with the actual stimulus values or the correct class labels for each trial as feedback. Thus, they did not receive any explicitly probabilistic feedback, nor were they explicitly trained to perform probabilistic inference.

We manipulated sensory reliability via gain variables g multiplying the responses of the input populations. In each task, networks were tested with a wide range of gains or gain combinations (in tasks with more than one input). To test the generalization capacity of the networks, we trained them with a limited range of gains or gain combinations, as well as with the full range of test gains or gain combinations. The latter unrestricted training regime is called the "All $\mathbf{g}$" condition, whereas the former limited training regime is called the "Restricted $\mathbf{g}$" condition in what follows. The specific gain ranges and gain combinations used in each task are indicated below (see *Methods*).

Generic neural networks trained with error-based learning rules implement probabilistic inference in standard psychophysical tasks. The performance of well-trained networks is shown in Figure 2 for both "All $\mathbf{g}$" and "Restricted $\mathbf{g}$" training conditions in all tasks (complete learning curves of the networks, i.e. performance as a function of the number of training trials, are shown in supplementary Figure S1). In categorical tasks, the output nodes of the networks approximate the posterior probabilities of the classes given the input responses. Theoretical guarantees ensure that this property holds under general conditions with a wide range of loss functions [16] (*Discussion*). In all continuous tasks as well as in two of the categorical tasks, we found strong trial-by-trial correlations between the input gain and the sparsity of hidden layer responses (Figure 5e), which in turn induce correlations between the posterior precision and the sparsity of hidden layer responses in the continuous tasks. This intriguing phenomenon will be explained below.

Random networks. We also considered an alternative architecture, in which the input-to-hidden layer weights and the biases of the hidden units were set randomly and left untrained (*Methods*). Such networks can be plausible models of some neural systems [7], [35]. Although the performance of random networks improved with training, they required significantly larger numbers of hidden units than the corresponding fully trained networks to achieve a similar level of performance and, in some cases (e.g. in Kalman filtering), failed to achieve error rates as low as the fully trained networks even with very large numbers of hidden units (supplementary Figure S2). There are good theoretical reasons for the inefficiency of random networks: the approximation error of neural networks with adjustable hidden units scales as $O(1/n)$ with n denoting the number of hidden units, whereas for networks with fixed hidden units, the scaling is much worse: $O(1/n^{2/d})$ where d is the dimensionality of the problem [1]. Because random networks were substantially more inefficient than fully trained networks, we exclusively consider the fully trained networks for the rest of this paper.

Alternative representations of stimulus reliability in the input populations. Thus far, we have assumed that sensory reliability has a purely multiplicative effect on the responses of input neurons. Although this assumption appears to hold for the effect of contrast in visual cortex [34], it is known to be violated in the case of motion coherence in random dot displays [10], [28], and is unlikely to hold in the general case. The importance of this observation is that the Poisson-like PPC approach proposed in [23] cannot easily handle cases where "nuisance variables" such as contrast or coherence do not have a purely multiplicative effect on neural responses. By contrast, our approach does not make any assumptions about the representation of stimulus reliability in the input populations. We demonstrated this in two cases: (i) cue combination with tuning functions of the form reported in [10], [28] where stimulus coherence affects both the gain and the baseline of the responses (supplementary Figure S3) and (ii) cue combination with tuning functions whose width depends on a nuisance parameter (supplementary Figure S4).

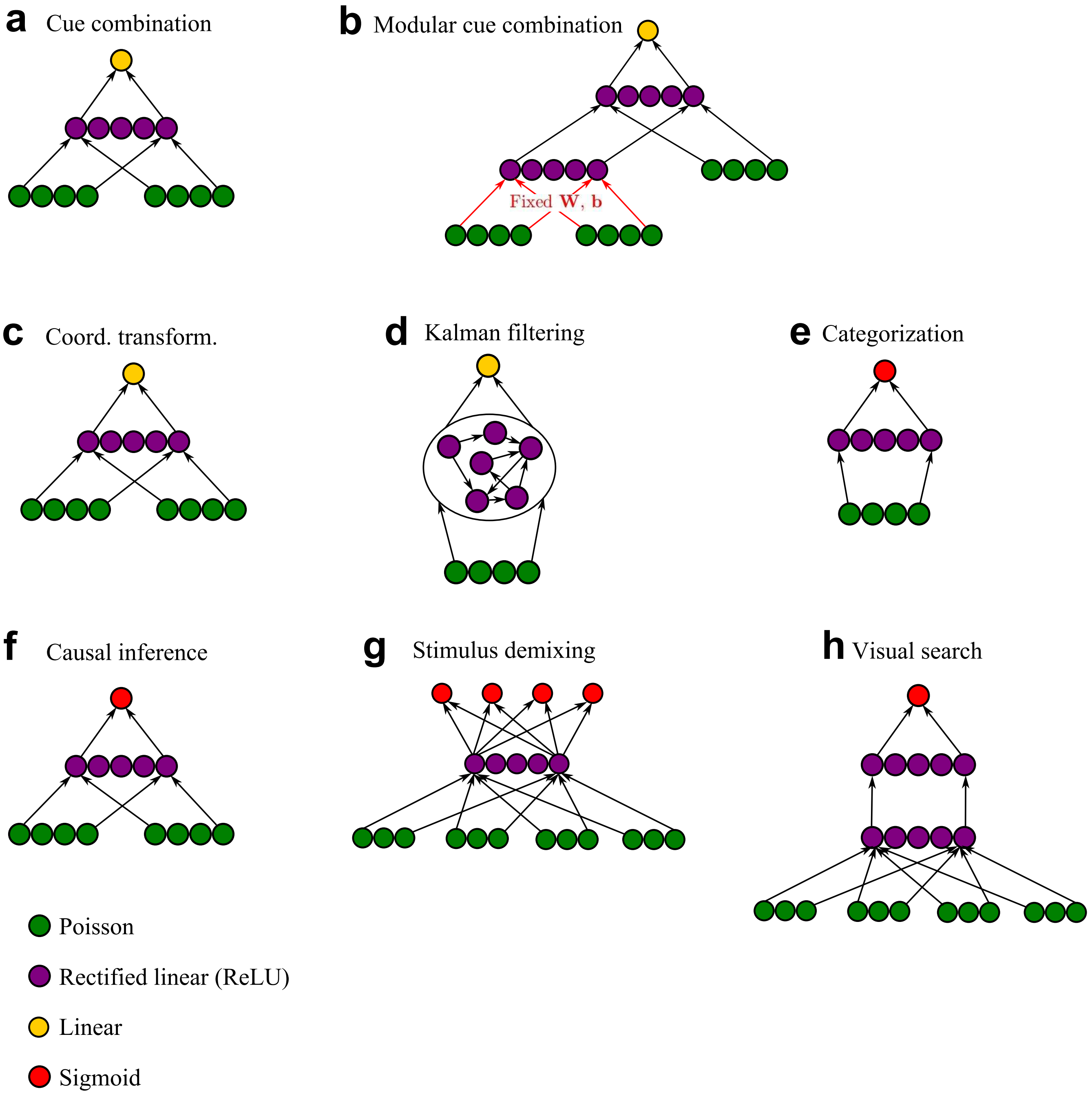

Figure 1: Network architectures used for the tasks considered in this paper. Different colors represent different types of units. For tasks with continuous output variables (a-d) linear output units were used, whereas for tasks with categorical output variables (e-h) sigmoidal output units were used.

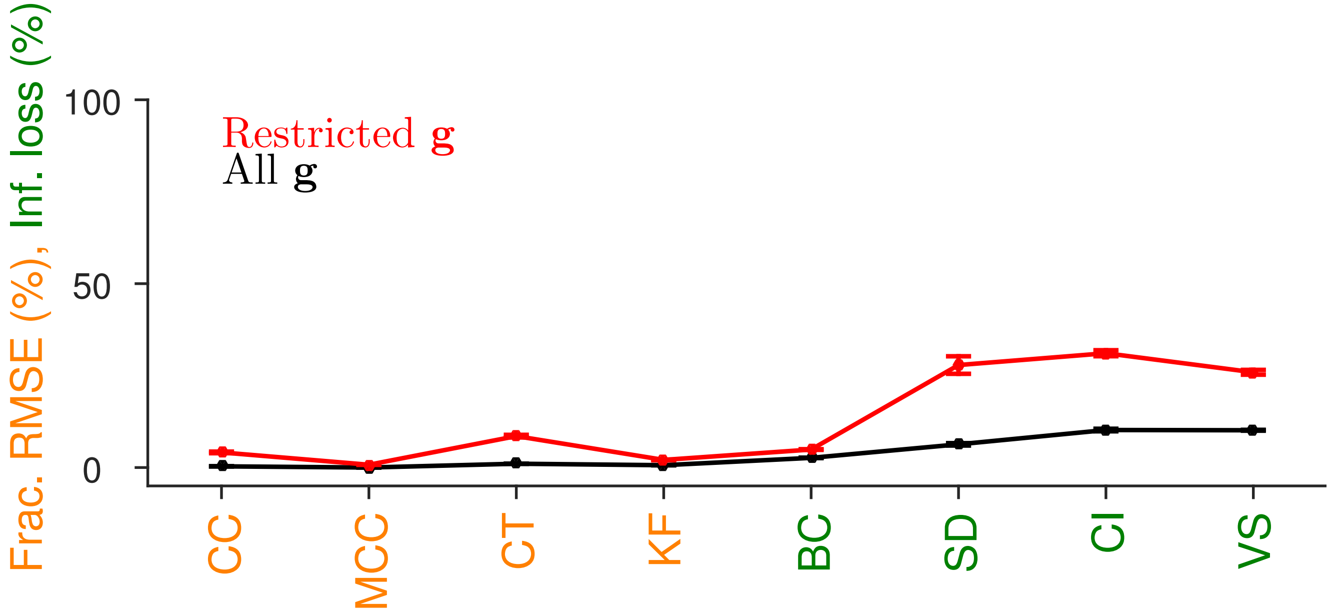

Figure 2: Performance of well-trained networks in all tasks. Error bars represent standard errors over 10 independent runs of the simulations. For continuous tasks, performance is measured in terms of fractional RMSE defined as $100 \times (\mathrm{RMSE}_{\mathrm{netw}} - \mathrm{RMSE}_{\mathrm{MLE}})/(\mathrm{RMSE}_0 - \mathrm{RMSE}_{\mathrm{MLE}})$ where $\mathrm{RMSE}_{\mathrm{netw}}$ is the root mean-squared error (RMSE) of the trained network, $\mathrm{RMSE}_{\mathrm{MLE}}$ is the RMSE of the maximum-likelihood estimator (MLE) and RMSE_0 represents the RMSE of a constant estimator that always outputs zero. With this measure, 0% corresponds to the performance of the MLE estimator, 100% corresponds to the performance of a constant estimator that always outputs zero, which approximately characterizes the behavior of an untrained network with small random, initial weights. For categorical tasks, performance is measured in terms of fractional information loss which is defined as the KL-divergence between the actual posterior and the network's output normalized by the mutual information between the class labels and the neural responses [4]. Abbreviations: CC (cue combination); MCC (modular cue combination); CT (coordinate transformation); KF (Kalman filtering); BC (binary categorization); SD (stimulus demixing); CI (causal inference); VS (visual search). Categorical tasks are labeled in green, continuous tasks in orange.

Thus, our networks can approximate optimal estimates regardless of the specific form in which stimulus reliability is encoded in the input populations, although the hidden layer representations that emerge from training can be different for different schemes (supplementary Figures S3-S4).

Modularity of the hidden layer representations. In the linear cue combination task, the representations learned by the hidden units were fully modular in the sense that the trained hidden units could be directly plugged into a network incorporating an additional input population without any re-training and with little information loss (Figure 1b and Figure 2). Unlike in [23], however, the format of variability differs between the layers.

Generalization to untrained stimulus conditions. As shown in Figure 2 (red line), the networks were able to generalize well beyond the training conditions in all tasks. An example is shown in Figure 3a for the cue combination task. In the "restricted $\mathbf{g}$" condition, we trained the networks with only two gain combinations, $\mathbf{g} = (g_1, g_2) = (5, 5)$ and $\mathbf{g} = (25, 25)$ and tested them on all gain combinations of the form (g_1, g_2) where $g_1, g_2 \in \{5, 10, 15, 20, 25\}$ with up to five-fold gain differences between the two input populations ("all $\mathbf{g}$" condition). To demonstrate that the trained networks performed qualitatively correct probabilistic inference, we set up cue conflict conditions (similar to the cue conflict conditions in psychophysical studies) where we presented slightly different stimuli to the two input populations and manipulated the degree of conflict between the cues. The network achieved low generalization error (fractional RMSE: 10.9%) even after as few as 50 trials in this impoverished training condition and performed qualitatively correct probabilistic inference in the untrained conditions. In particular, the network correctly adjusted the weights assigned to the two cues as their reliability varied (Figure 3a).

The successful generalization performance of the neural networks is a result of two factors. First, the target function is approximately invariant to gain manipulations that differ between the training and test conditions. In cue combination, for instance, the target function (Equation 7) is approximately invariant to the scaling of the input populations by arbitrary gains g_1 and g_2. The second factor is the network's inductive biases, i.e. how it tends to behave outside the training domain. These inductive biases depend on the details of the network architecture [30]. The importance of the network architecture in the extrapolation behavior of neural networks can be illustrated with a simple example (Figure 3b-c). In this example, we trained two neural networks with different hidden unit activation functions on the same two-dimensional regression problem that involved learning the same output value for two clusters of widely separated training points. The training data are indicated by the red dots in Figure 3b-c. The network with rectified linear activation functions learned an approximately constant function (Figure 3b), whereas the network with cosine activation functions learned a more complex function outside the training domain (Figure 3c), even though both networks performed equally well on the training data.

Sparsity-based representation of sensory reliability. We now explain the observed correlations between the input gain and the sparsity of hidden layer responses in the trained networks. We first note that in the tasks with continuous output variables, the optimal solution is invariant to a scaling g of the input responses (*Methods*). We then asked how the hidden unit responses should behave to make the neural network's outputs invariant to such scalings. To answer this question, we first derived an analytical expression for the mean response of a typical hidden unit $\bar{\mu}$, as a function of the input gain g, the mean input μ to the hidden unit for unit gain, and the mean μ_b and the standard deviation σ_b of the biases of the hidden units (*Methods*). To minimize the dependence of the mean hidden unit response on g, we introduced a measure of the total sensitivity of $\bar{\mu}$ to variations in g called T_{var} (*Methods*; Equation 3) and minimized T_{var} with respect to μ, μ_b and σ_b. T_{var} was minimized for slightly negative mean input μ and large μ_b and σ_b values (Figure 4a). The negativity of the mean input μ implies that as the gain g increases, the distribution of the total input h to the unit shifts to the left, causing a larger proportion of the distribution to remain below the threshold, hence decreasing the probability that the neuron will have a non-zero response (Figure 4c). This mechanism underlies the sparsification of the hidden unit responses with increasing g. Because increasing g also increases the variance of h, the mean response for those inputs that do cross the threshold increases (Figure 4d). As a result, the mean response of the neuron, which is a product of these two terms, remains roughly constant (Figure 4e). If μ is larger, on the other hand, the probability of non-zero responses as a function of g assumes a flatter profile, causing the mean response to increase with g.

Because the input responses are always non-negative, the only way μ can be negative is if the mean input-to-hidden layer weight is negative. For a simple gain-independent orientation estimation task, we indeed found this to be case (Figure 4f, first column). Consequently, as predicted from our simplified statistical description, we found a strong positive correlation between the sparsity of hidden layer responses and the mean input response (Figure 4f, last column), but only a weak, slightly negative correlation between the mean hidden layer response and the mean input response (Figure 4f, fourth column). Moreover, the mean bias of the hidden units, μ_b, was positive (Figure 4f,

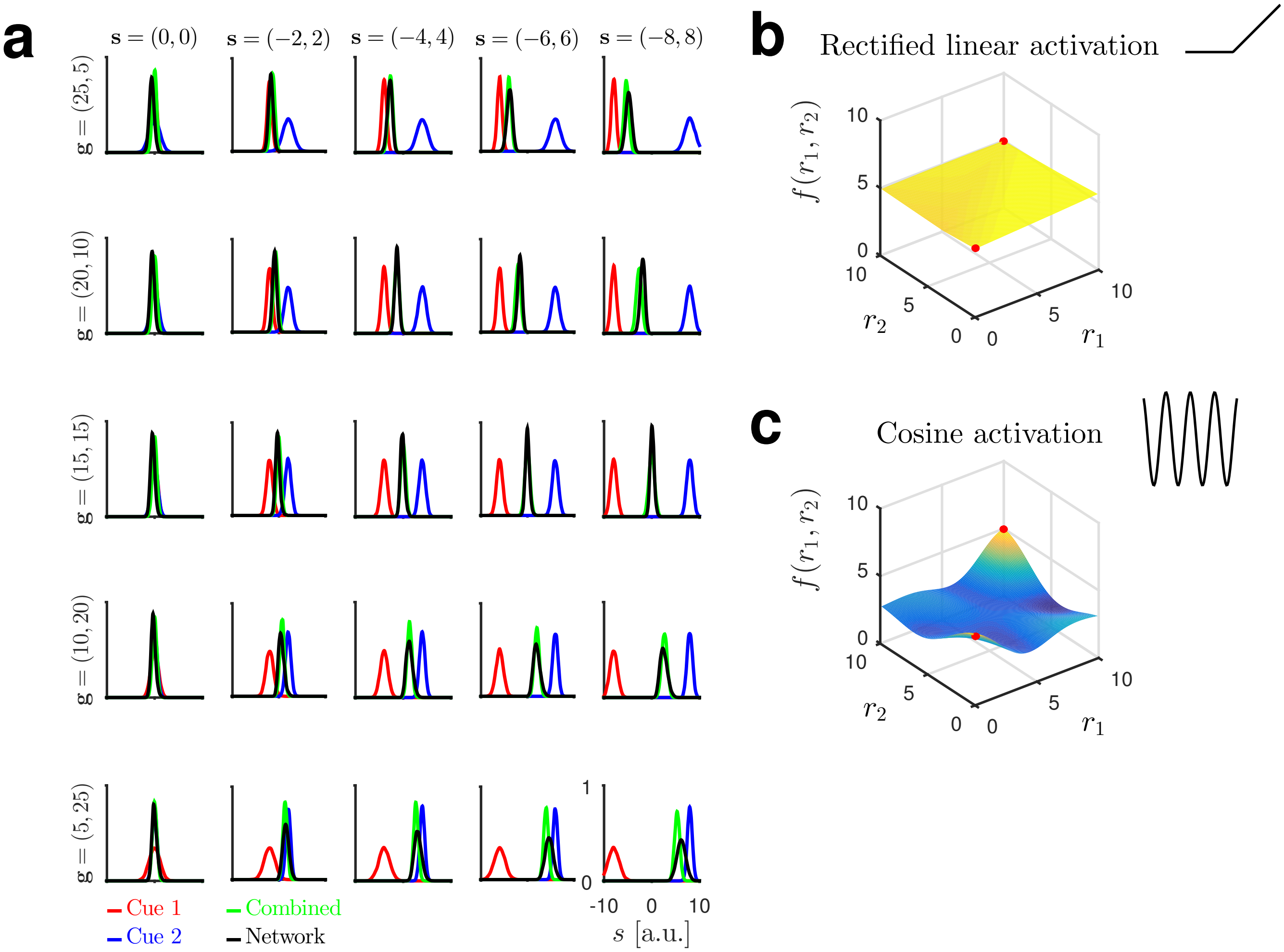

Figure 3: Generalization capacity of neural networks. (a) Optimal and the network's estimate distributions in cue conflict conditions. Note that the network was trained with only 50 trials in the restricted **g** condition, hence it did not see any of the conditions shown here during training. (b-c) Demonstration of the importance of network architecture, in particular the choice of hidden unit activation function, in the extrapolation behavior of neural networks. Training data are indicated by the red dots.

second column), again consistent with the result from the simplified model above. Finally, we also found a strong positive correlation between the biases of the hidden units and the magnitude of their output weights (Figure 4f, third column) which can be explained through a simple signal-to-noise ratio argument (*Methods*).

The same type of analysis applies to the categorical tasks as well. However, the difference is that for some of our tasks with categorical outputs, in the optimal solution, the net input to the output unit had a strong dependence on g. For example, in causal inference, the input to the sigmoidal output unit scales approximately linearly with g (Equation 12). Similarly, in visual search, both global and local log-likelihood ratios have a strong dependence on g (through $\mathbf{r}_i$ in Equations 18-19). In binary categorization, however, the net input to the sigmoidal output unit should be approximately invariant to a scaling g of the input responses $\mathbf{r}$ (Equation 11).

We therefore predicted that for those tasks where the net input to the output unit is approximately g-invariant, there should be a positive correlation between the sparsity of hidden unit responses and the input gain and no (or only a weak) correlation between the mean hidden unit response and the input gain. On the other hand, in tasks such as causal inference, where the net input to the output unit has a strong g-dependence, we predicted a positive correlation between the mean hidden unit response and the input gain and no (or only a weak) correlation between the sparsity of hidden unit responses and the input gain. We tested these predictions on our trained networks and found that they were indeed borne out (Figure 5d-e). For causal inference and visual search tasks, the correlation between the mean input and the sparsity of hidden layer responses was weak (Figure 5e), whereas for the remaining tasks, it was strong and positive. The opposite pattern was seen for the correlation between the mean input and the mean hidden layer response (Figure 5d). The sparsity-based representation of the input gain was again driven by the negativity of the mean input-to-hidden layer weights (Figure 5a).

The difference between these two types of tasks (g-invariant and g-dependent) was also reflected in the tuning functions that developed in the hidden layer of the networks. For g-invariant (or approximately g-invariant) tasks such as coordinate transformation, the input gain g sharpens the tuning of the hidden units (Figure 6b), whereas for gain-dependent tasks, it acts more like a multiplicative factor scaling the tuning function without changing its shape (Figure 6d).

Insufficient-training hypothesis for suboptimal probabilistic inference. The dependence of the networks' performance on the number of training trials (Figure S1) suggests a possible explanation for deviations from optimality sometimes observed in experimental studies: i.e. insufficient training in the task. Testing this hypothesis rigorously is complicated by possible prior exposure of the subjects to similar stimuli or tasks under natural conditions. Among the tasks considered in this paper, the binary categorization task minimizes such concerns, because it involves classifying stimuli into arbitrary categories. Moreover, in this task, the behavior of both human and monkey observers were best explained by heuristic models that were quantitatively suboptimal, but qualitatively consistent with the optimal model [32]. Therefore, we sought to test the insufficient-training hypothesis for suboptimality in this task. The structure of the categories and the decision boundaries predicted by the optimal (OPT) and three suboptimal models are shown in Figure 7a-b. The learning curve of a monkey subject is shown in Figure 7 together with the performance of a neural network that received the same sequence of trials as the subject. The input noise of the network was matched to the sensory noise estimated for the subject and the learning rate of the network was optimized to fit the learning curve of the subject (*Methods*). Besides providing a good fit to the learning curve of the subject, the neural network also correctly predicted the model that best fit the subject's data, i.e. the suboptimal LIN model. When we performed the same type of analysis on human subjects' data, human observers consistently outperformed the networks and the networks failed to reproduce the learning curves of the subjects (Figure 7e-g). There might be several possible non-exclusive explanations for this finding. First, prior to the experiment, human observers were told about the task, including what examples from each category looked like. This type of knowledge would be difficult to capture with error-based learning alone and might have given human observers a “head-start” in the task. Second, human observers might have benefited from possible prior familiarity with similar tasks or stimuli. Third, human observers might be endowed with more powerful computational architectures than simple generic neural networks that allow them to learn faster and generalize better [13].

Low computational complexity of standard psychophysical tasks and the efficiency of generic networks. Figure 8 shows the performance of the fully trained networks as a function of the number of hidden units in all tasks when the networks were trained with large numbers of training trials. In all tasks, performance asymptoted at relatively small numbers of hidden units, usually on the order of tens of hidden units. This result can be explained by two factors: (i) the low-dimensional nature of the solutions and (ii) the smoothness of the target functions in those low-dimensional spaces. The first property arises because, although the inputs are high

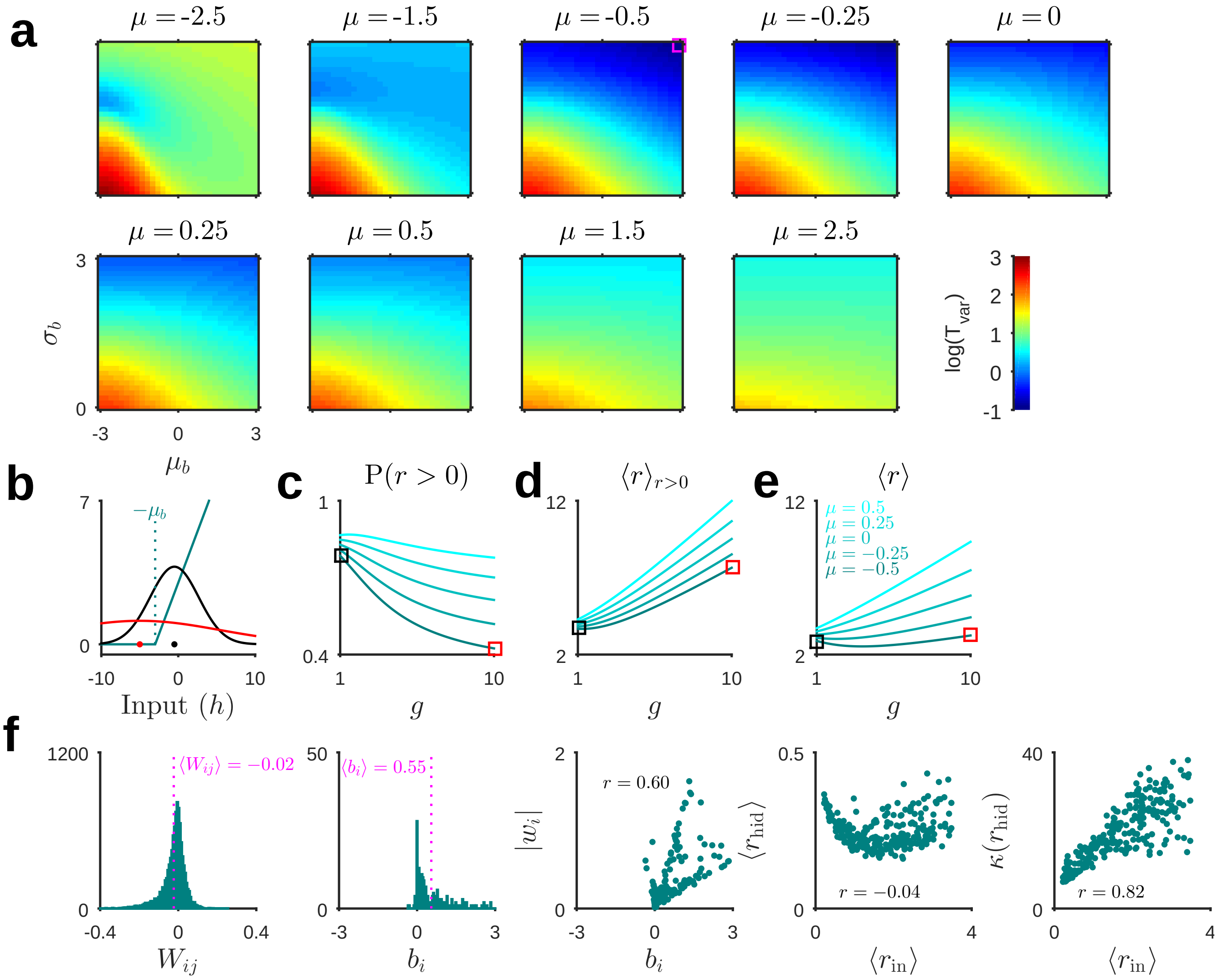

Figure 4: The mechanism underlying the sparsity-based representation of sensory reliability. (a) The variability index T_{var} (plotted in log units) as a function of the parameters μ, μ_b and σ_b. (b) The mean bias of the hidden units, μ_b, is slightly positive, or equivalently their mean threshold is slightly negative, and the mean input μ is slightly negative. Increasing the gain g thus shifts the input distribution to the left: compare the black and red lines for input distributions with a small and a large gain, respectively. The means of the input distributions are indicated by small dots. (c) This, in turn, decreases the probability of non-zero responses, but (d) increases the mean of the non-zero responses; hence (e) the mean response of the hidden units, being a product of the two, stays approximately constant as the gain is varied. (f) For a network trained in a simple gain-invariant stimulus estimation task, distributions of the input-to-hidden layer weights, W_{ij}; the biases of the hidden units, b_i; scatter plot of the biases vs. the magnitude of the output weights for the hidden units; scatter plot of the mean input $\langle r_{\text{in}} \rangle$ vs. the mean hidden unit activity $\langle r_{\text{hid}} \rangle$; as well as the scatter plot of the mean input vs. the kurtosis of hidden unit activity $\kappa(r_{\text{hid}})$.

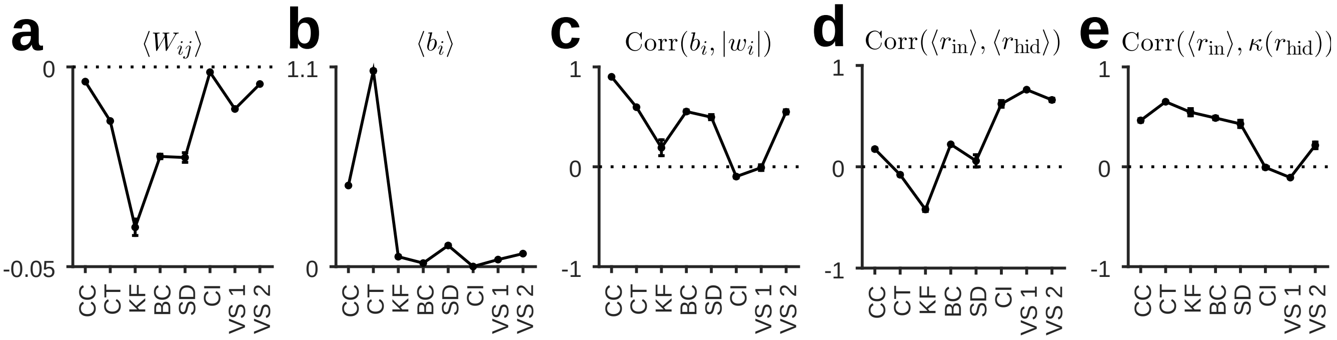

Figure 5: Parameter statistics for the trained networks. CC: cue combination; CT: coordinate transformation; KF: Kalman filtering; BC: binary categorization; SD: stimulus demixing; CI: causal inference; VS 1: visual search (first hidden layer); VS 2: visual search (second hidden layer).

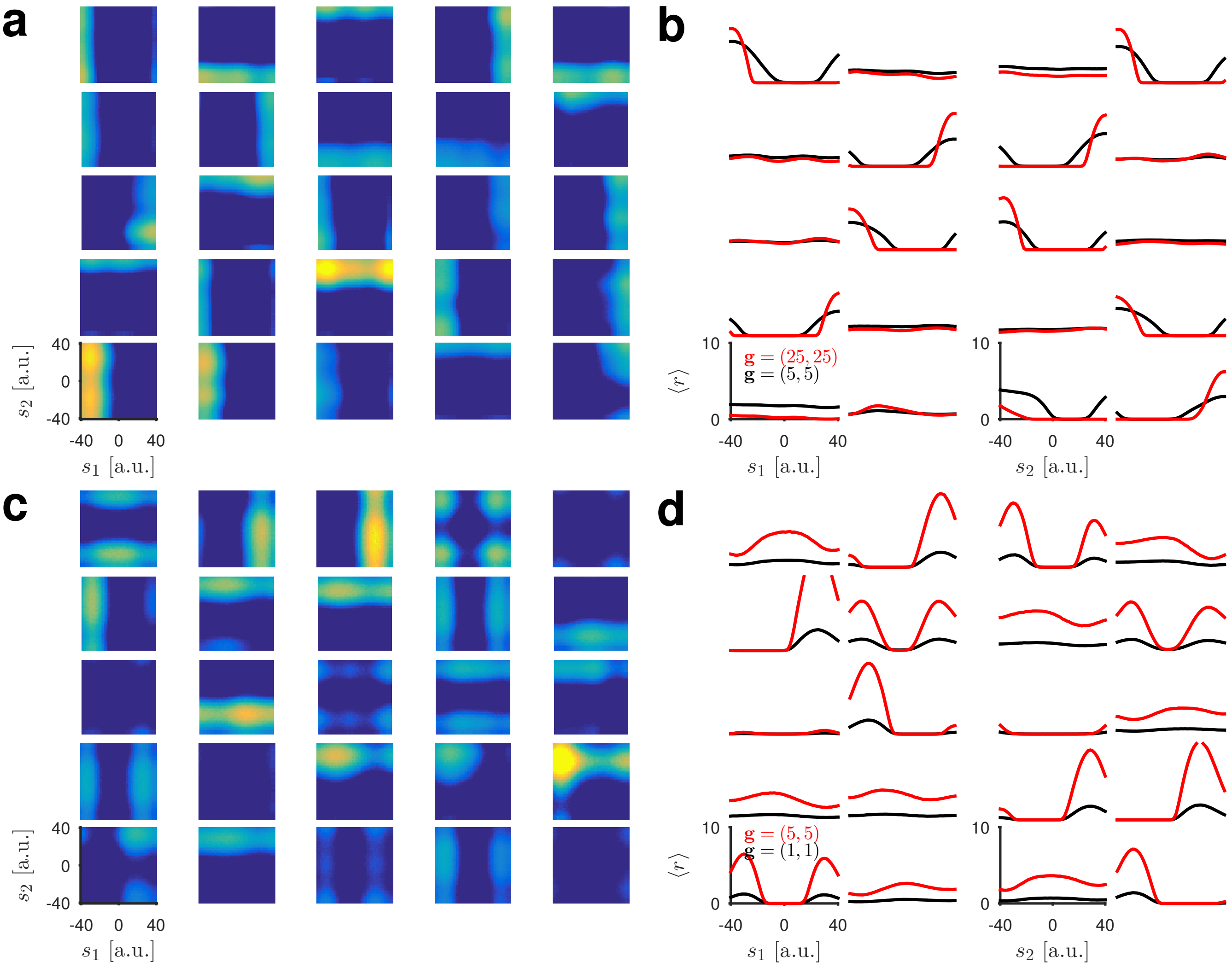

Figure 6: Hidden layer tuning functions for well-trained networks in the coordinate transformation (a-b) and causal inference tasks (c-d). (a) Two-dimensional tuning functions of the top 25 hidden units in a network trained in the coordinate transformation task ($\mathbf{g} = (25, 25)$). (b) One-dimensional tuning functions of the top 10 hidden units at two input gains. (c-d) Similar to (a-b), but for a network trained in the causal inference task. In (c), $\mathbf{g} = (5, 5)$.

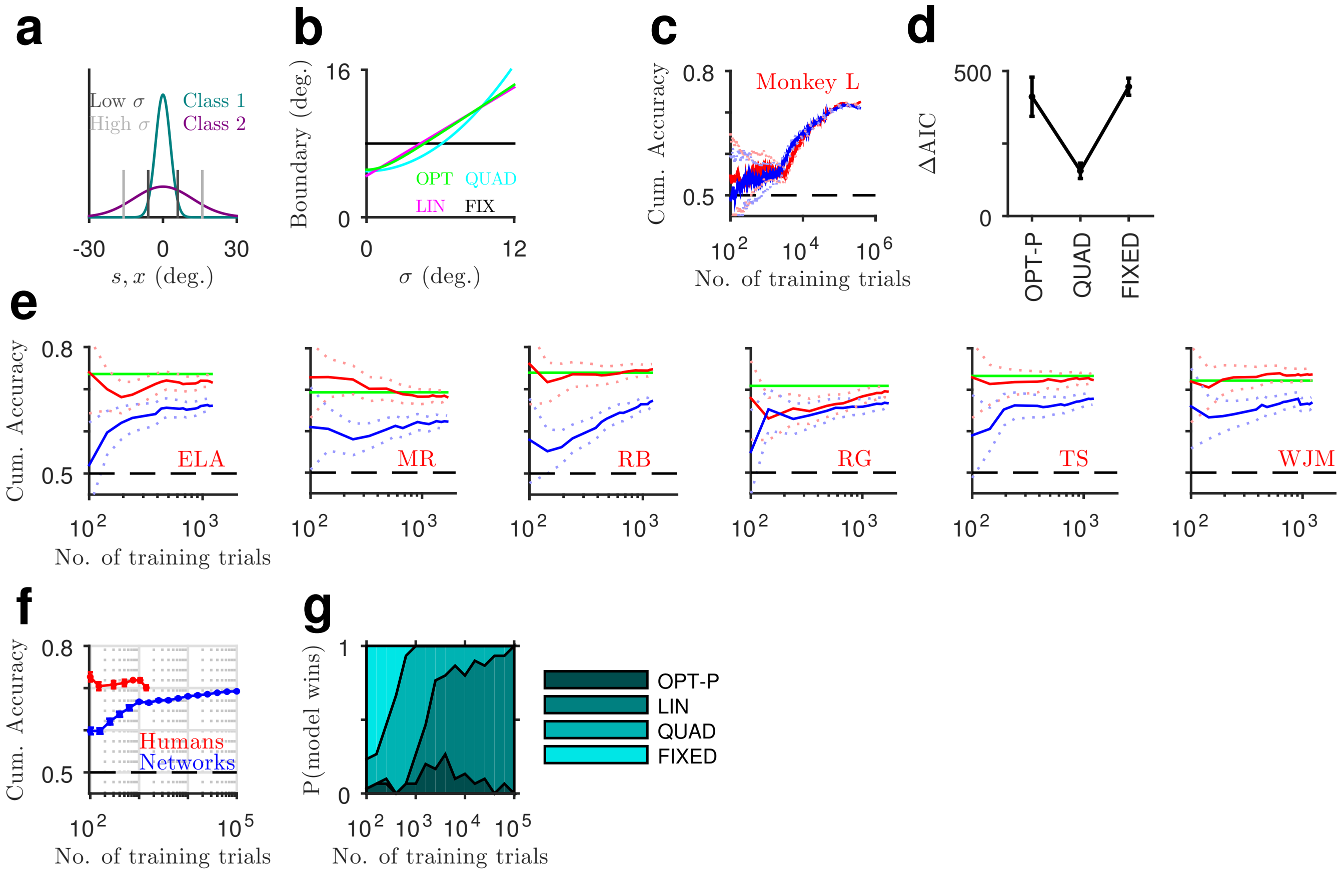

Figure 7: Error-based learning in generic neural networks can explain a monkey subject's performance, but not human subjects' performance in a probabilistic binary categorization task involving arbitrary categories. (a) Structure of the two categories. Vertical lines indicate the optimal decision boundaries at low and high noise, σ. (b) Dependence of the decision boundary on sensory noise, σ, for the optimal model (OPT) and three suboptimal heuristic models. (c) Cumulative accuracy of monkey L (red) compared with the cumulative accuracy of a neural network trained with the same set of stimuli (blue). The network was trained fully on-line. The input noise in the network was matched to the sensory noise estimated for the monkey and the learning rate was optimized to match the monkey's learning curve (*Methods*). Dotted lines indicate the 95% binomial confidence intervals. (d) Model comparison results for 30 neural networks trained with the same set of input noise and learning rate parameters as the one shown in (c). OPT-P is equivalent to an OPT model where the prior probabilities of the categories are allowed to be different from 0.5 (in the experiment, both categories were equally likely). Error bars represent standard errors over 30 networks. The heuristic LIN model had the best AIC score for 29 networks, OPT-P had the best AIC score for the remaining network. AIC scores are reported relative to the AIC score of the LIN model. Lower scores mean better fits. (e) Performance of the human subjects over the course of the experiment (red) compared with the performance of noise-matched neural networks (blue) that received the same set of trials as the humans. Green lines indicate the best achievable accuracy given the sensory noise estimated for the subject. (f) Average performance of human subjects (red), average performance of 30 neural networks whose input noise was set to the mean sensory noise estimated for the human subjects (blue). (g) Fractions of winning models, as measured by their AIC scores, over the 30 neural network subjects shown in blue in (f). Early on in the training, FIXED is the most likely model to win. QUAD and then LIN become the most likely models as training progresses.

dimensional, the solutions can be expressed as simple algebraic functions of a small number of projections of the inputs onto one-dimensional subspaces (e.g. see Equation 7). Theoretical results suggest that the second property also plays a role, because in generic networks, the optimal number of hidden units is expected to be smaller for more smooth target functions [1]. The efficiency of our generic networks contrasts sharply with the efficiency of the manually crafted networks in earlier PPC studies [3] [4], [23], [24], [25], [32]: for most tasks these hand-crafted networks used a quadratic expansion which requires $O(n^2)$ hidden units for n input units. This would, for example, require on the order of 2500 hidden units in the coordinate transformation task ($n = 50$).

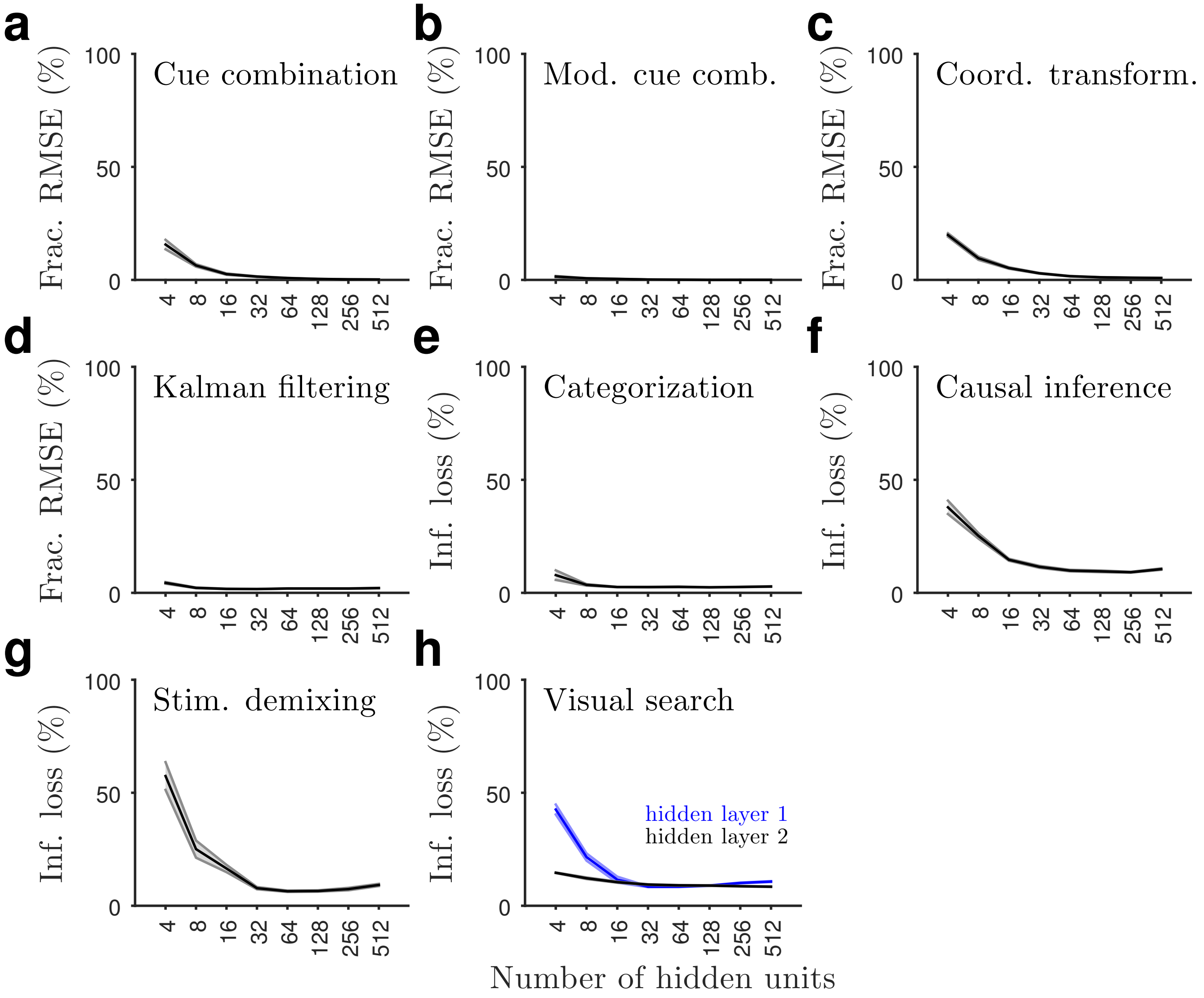

Figure 8: Low computational complexity of standard psychophysical tasks and the efficiency of generic networks. Performance is plotted as a function of the number of hidden units for the eight tasks considered in this paper. For the visual search task, the number of hidden units in the two layers was varied separately: when varying the number of hidden units in the second layer, the number of hidden units in the first layer was set to 128; when varying the number of hidden units in the first layer, the number of hidden units in the second layer was set to 64.

Discussion

We showed that very small generic neural networks trained with a standard error-based learning rule, but without any explicitly probabilistic feedback or training objective, implement probabilistic inference in simple probabilistic psychophysical tasks and generalize successfully well beyond the training data they receive.

An important question to consider is why generic neural networks work as well as they do in probabilistic inference problems. For tasks with continuous outputs, our networks were trained to minimize the squared error loss function, which is minimized by the posterior mean estimate. Given the universal approximation guarantees for multilayer neural networks with rectified linear hidden units [21], it is not surprising that our networks can approximate the posterior mean, a sufficient condition for performing probabilistic inference, given enough hidden units and training data. However, the findings that near-optimal performance can be achieved even in small networks trained with a relatively small number of training examples and that the networks can generalize successfully beyond the training data they receive depend on the particular problems we studied, in particular, on their low-dimensional nature and their smoothness properties, hence are not predicted by the universal approximation results. The networks can, in principle, be trained to minimize loss functions other than squared error, in which case they would approximate alternative Bayes estimators.

For tasks with categorical outputs, even better theoretical results hold. In this case, under mild conditions, the output layer of a multilayer neural network is guaranteed to converge to the posterior probabilities of the classes with any "reasonable" loss function, where "reasonable" has a formal definition according to which all commonly used loss functions such as cross-entropy or mean squared error are reasonable [16].

Relationship to previous works. Our work is consistent with the probabilistic population coding framework according to which, by virtue of the variability of their responses, neural populations encode probability distributions rather than single estimates and computations with probability distributions can be carried out by suitable operations on the corresponding neural responses [23], [40]. However, our work disagrees with the existing literature on the implementation of such computations. We show that these computations do not require any special neural operations or network architectures than the very generic ones that researchers have been using for decades in the neural network community [33], [37], [41].

The starting point of the recent literature on probabilistic population coding has been the principle that Poisson-like neural variability be preserved between the input and output of a network, because this leads to a fully modular code that can be decoded with the same type of decoder throughout the network [23]. To obtain actual networks, these studies then postulated a literal, one-to-one correspondence between the required neural computations that must be computed at the population level and the computations individual neurons perform. This literal interpretation has led to inefficient neural architectures containing intermediate neurons that are artificially restricted to summing or multiplying the activities of at most two input neurons and that perform substantially different operations in different tasks (e.g. linear summation, multiplication or different forms of divisive normalization) [3] [4], [23], [24], [25], [32].

Our generic networks are not necessarily inconsistent with the principle of the preservation of Poisson-like variability between the input and output of a network. Our categorical networks already satisfy this principle, and our continuous networks satisfy it if, instead of a purely linear decoder, we use a linear decoder that is normalized by the total activity in the hidden layer (supplementary Figure S5). However, our results show that it is unnecessary and inefficient to postulate a direct correspondence between population-level and individual-neuron computations: standard neural networks with rectified linear hidden units that perform the same type of operation independent of the task implement population-level computations required for optimal probabilistic inference far more efficiently.

Generic neural networks of the type we used in this paper have been used previously to explain neural responses in prefrontal cortex in a context-dependent decision making task [26], neural responses to naturalistic visual stimuli in higher visual areas [39] and responses of neurons in posterior parietal cortex performing a type of coordinate transformation [41]. Our results suggest that similar networks can be plausible neural models in tasks that require probabilistic computations.

Experimental predictions. Our results lead to several experimentally testable predictions. First, for gain-invariant tasks, we predict a novel sparsity-based coding of sensory reliability in cortical areas close to the readout. Stimulus manipulations that increase sensory reliability such as an increase in contrast or coherence of the stimulus would be expected to increase the sparseness of the population activity in these areas. Another straightforward consequence of this relationship would be a positive correlation between the performance of the animal in the task and the population sparsity of neurons recorded from the same areas. Second, for tasks such as causal inference, we predict a different coding of sensory reliability based on the mean activity in areas close to the readout. Moreover, based on our simplified model of the mechanism underlying these two types of codes, we expect a trade-off between them: the stronger the correlation between sparsity and sensory reliability, the weaker the relationship between the mean activity and sensory reliability and vice versa. This can be tested with population recordings from multiple areas in multiple tasks. Third, at the level of single cells, we predict tuning curve sharpening with increased input gain in tasks where a sparsity-based coding of reliability is predicted (Figure 6a-b). Such tuning curve sharpening has indeed been observed in cortical areas MT [6], MST [17] and MSTd [28]. On the other hand, we expect the input gain to act more like a multiplicative factor in tasks where a mean activity based coding of reliability is predicted (Figure 6c-d).

Sparse and reliable neural responses have been observed under natural stimulation conditions [8], [15], [36]. Inhibitory currents have been shown to be crucial in generating such sparse and reliable responses [8], [14], analogous to the importance of the mean input to a hidden unit being negative in our simplified model of the sparsity-based coding of sensory reliability (Figure 4b).

Limitations. Our networks are highly idealized models of real neural circuits. First, except for the recurrent network that was used in the Kalman filtering task, they are all feed-forward networks, whereas cortical networks are usually highly recurrent. However, networks with feedback connections can sometimes behave effectively like a feedforward network [12], [29]. Feedforward networks also currently provide the best characterization of the neural responses in higher visual cortical areas [39], even though these areas are known to involve abundant feedback connections both within the same area and between different areas. Therefore, we expect that insights gained from understanding feedforward networks will not be completely irrelevant for understanding real cortical circuits. Second, our networks do not respect Dale's law, which states that the synaptic projections of a neuron can be either all inhibitory or all excitatory, but not mixed. However, there are relatively simple ways of turning a generic network that does not respect Dale's law into an equivalent one that does [31]. Thus, we do not expect this deviation from biological reality to be consequential either. Third, our networks were trained with the backpropagation algorithm, which is usually considered to be biologically unrealistic due to its non-locality. Although the backpropagation algorithm in its standard form we have implemented is indeed biologically unrealistic, biologically plausible approximations to backpropagation have been put forward recently [5], [22]. Therefore, it is quite likely that one need not compromise the power of backpropagation in order to attain biologically plausibility.

Finally, as exemplified by the inability of error-based learning to account for the performance of human observers in the binary categorization task, we do not expect error-based learning in generic neural networks to fully account for all aspects of the performance of human observers, and possibly non-human observers as well, even in simple tasks. Relatively mundane manipulations such as changing the target or the distractors, or the number of distractors in a visual search task, changing the duration of a delay interval in a short-term memory task require wholesale re-training of generic neural networks, which seems to be inconsistent with the way human observers, and possibly non-human observers, can effortlessly generalize over such variables. More powerful architectures that combine a neural network controller with an external memory can both learn faster and generalize better [13], offering a promising direction for modeling the generalization patterns of observers in simple psychophysical tasks.

Methods

Neural networks: In all networks, the input units were independent Poisson neurons: $\mathbf{r}_{in} \sim \text{Poisson}(\mathbf{f}(s, c))$ where $\mathbf{f}$ is the vector of mean responses (tuning functions), s is the stimulus and c is a stimulus contrast or coherence variable that controls the quality of sensory information. For most of the results in the paper, we assume that the effect of c can be described as a multiplicative gain scaling: $\mathbf{f}(s, c) = g(c)\mathbf{f}(s)$ where the individual tuning functions comprising $\mathbf{f}(s)$ were either linear, Gaussian or von Mises in different tasks. For the cue combination task, we also considered tuning functions where c did not have a purely multiplicative effect, but affected the baseline responses as well [10]: $f(s, c) = cf(s) + (1 - c)\beta$ with $0 \leq c \leq 1$ where β was chosen such that the mean response of the input population was independent of c.

The hidden units in both feed-forward and recurrent networks were rectified linear units. In feed-forward networks, the hidden unit responses are described by the equation: $\mathbf{r}_{hid} = [\mathbf{W}_{in}\mathbf{r}_{in} + \mathbf{b}]_+$ and in recurrent networks by the equation: $\mathbf{r}_{hid,t+1} = [\mathbf{W}_{in}\mathbf{r}_{in,t+1} + \mathbf{W}_{rec}\mathbf{r}_{hid,t} + \mathbf{b}]_+$, where $\mathbf{W}_{in}$ and $\mathbf{W}_{rec}$ are the input and recurrent weights respectively and $[\cdot]_+$ denotes elementwise rectification. For tasks with continuous output variables, the network output corresponds to a linear combination of the hidden unit responses: $y = \mathbf{w}^\top \mathbf{r}_{hid}$, and in tasks with categorical variables, the network output was given by a linear combination of the hidden unit responses passed through a sigmoid nonlinearity: $y = \sigma(\mathbf{w}^\top \mathbf{r}_{hid})$. For the stimulus demixing task where there were more than a single output variable, the outputs were softmax units instead.

In the main simulations, feed-forward networks with a single hidden layer had 200 hidden units, the two-hidden layer feed-forward network for the visual search task had 130 and 70 hidden units in its first and second hidden layers, and the recurrent network for the Kalman filtering task had 30 recurrent units. Partially random networks had 10 times the number of hidden units in the corresponding fully trained network. In cue combination, modular cue combination, coordinate transformation, Kalman filtering, and causal inference tasks, there were 50 input neurons per input population. To make our results comparable to earlier results, we used 20 input neurons per input population in the binary categorization and visual search tasks and 10 input neurons per input population in the stimulus demixing task.

Training procedure: The feed-forward networks were trained with the standard back-propagation algorithm [33]. The recurrent networks for the Kalman filtering task were trained with back-propagation through time [37]. The algorithms were implemented with stochastic gradient descent with a momentum term. The batch size for the updates was 10. The learning rate was decreased over the course of training according to $\eta_0/(1+\gamma t)$ where $t = 0, 1, \ldots, N_e - 1$ refers to the epoch number. The number of training epochs, N_e, was fixed at 100 and the learning rate parameters η_0, γ, as well as the momentum coefficient were optimized with grid searches. The weights of the units in the network were initialized to small random values drawn from a zero-mean Gaussian distribution with a standard deviation of 0.01. In some tasks, we found it necessary to initialize the biases of the hidden units to large values; therefore, the biases of the hidden units were initialized to random values drawn from a zero-mean Gaussian distribution with standard deviation σ_b, where σ_b was optimized using a grid search.

Training conditions: The "all **g**" conditions in different tasks were as follows. In cue combination and coordinate transformation tasks, all 25 pairs of the form (g_1, g_2) with $g_1, g_2 \in \{5, 10, 15, 20, 25\}$ were presented an equal number of times. In Kalman filtering, g was uniformly drawn between 0.3 and 3 at each time step. In binary categorization, the six gain values, $g \in \{0.37, 0.9, 1.81, 2.82, 3.57, 4\}$, were presented an equal number of times. These gain values were calculated from the mean noise parameter values reported for the human subjects in [32]. In causal inference, all 25 pairs of the form (g_1, g_2) with $g_1, g_2 \in \{1, 2, 3, 4, 5\}$ were presented an equal number of times. In stimulus demixing, following [4], c was uniformly and independently drawn between 2 and 9 for each source. In visual search, g was randomly and independently set to either 0.5 or to 3 for each stimulus.

The "restricted **g**" conditions in different tasks were as follows. In cue combination and coordinate transformation tasks, the two pairs $(g_1, g_2) \in \{(5, 5), (25, 25)\}$ were presented an equal number of times. In Kalman filtering, g was randomly and independently set to either 1 or to 2 at each time step. In binary categorization, g was always 4.2. This gain value corresponds to 100% contrast as calculated from the mean noise parameter values for the human subjects reported in [32]. In causal inference, all three pairs of the form $(g_1, g_2) \in \{(2, 4), (3, 3), (4, 2)\}$ were presented an equal number of times. In stimulus demixing, c was either set to 2 for all sources or else set to 9 for all sources. Similarly, in visual search, g was either set to 0.5 for all stimuli or else set to 3 for all stimuli.

Mean-field model of hidden unit responses: For a given input activity $\mathbf{r}$, we consider the responses of the hidden units as realizations of a random variable r_{hid}. The output weights are also assumed to be realizations of a random variable w. We further assume that w and r_{hid} are independent. The network's output is then proportional to $\langle w \rangle \langle r_{hid} \rangle$. We want to make this expression invariant to the input gain g. We first introduce a measure of the total sensitivity of this expression to variations in g. We will do this by computing the magnitude of the derivative of $\langle w \rangle \langle r_{hid} \rangle$ with respect to g and integrating over a range of g values, but we first switch to the log domain to remove the dependence of the solution on the arbitrary scale of the outputs and the output weights. Then, we can ignore the output weights (which are already gain invariant) and only consider $\log\langle r_{hid} \rangle$. We now have to find an expression for $\langle r_{hid} \rangle$. The net input to a typical hidden unit is given by:

$$g\mathbf{w}_{in}^\top \mathbf{r} + b \sim \mathcal{N}(\mu_* \equiv g\mu + \mu_b, \sigma_*^2 \equiv g^2\sigma^2 + \sigma_b^2) \tag{1}$$

where $\mathbf{w}_{in}$ are the input weights to a typical hidden unit. Then:

$$\bar{\mu} \equiv \langle r_{hid} \rangle = \langle [g\mathbf{w}_{in}^\top \mathbf{r} + b]_+ \rangle = \Big[1 - \Phi\Big(\frac{-\mu_*}{\sigma_*}\Big)\Big]\mu_* + \phi\Big(\frac{-\mu_*}{\sigma_*}\Big)\sigma_* \tag{2}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the cdf and the pdf of the standard Gaussian distribution. As mentioned above, we then introduce the following measure of the total sensitivity of $\bar{\mu}$ to variations in g:

$$T_{\text{var}} = \int_{g_{min}}^{g_{max}} |(\log \bar{\mu}(g))'| dg = \int_{g_{min}}^{g_{max}} \Big|\frac{\bar{\mu}'(g)}{\bar{\mu}(g)}\Big| dg \tag{3}$$

where the prime represents the derivative with respect to g. We set $\sigma = 1$ and then minimized T_{var} numerically with respect to μ, μ_b and σ_b.

Signal-to-noise ratio calculation: The variance of a typical hidden unit can be calculated analytically:

$$\text{Var}([g\mathbf{w}_i^\top \mathbf{r} + b]_+) = \bar{\mu}\mu_* + \frac{\sigma_*^2}{1 - \Phi\Big(\frac{-\mu_*}{\sigma_*}\Big)} - \bar{\mu}^2 \tag{4}$$

The inverse of the signal-to-noise ratio, defined as the variance over mean of the hidden unit activity is given by:

$$\begin{aligned} \text{SNR}^{-1} &= \mu_* + \frac{\sigma_*^2}{\bar{\mu}\Big(1 - \Phi\Big(\frac{-\mu_*}{\sigma_*}\Big)\Big)} - \bar{\mu} && (5) \\ &= -\Big[\Big(1 - \Phi\Big(\frac{g\mu + b}{g\sigma}\Big)\Big)(-g\mu - b) + g\sigma\phi\Big(\frac{g\mu + b}{g\sigma}\Big)\Big] + \frac{g\sigma\sigma_*}{\bar{\mu}\Big(1 - \Phi\Big(\frac{-\mu_*}{\sigma_*}\Big)\Big)} && (6) \end{aligned}$$

After taking the derivative with respect to b, a straightforward analysis shows that both terms in the last equation are decreasing functions of b.

Task details: In the linear cue combination task, the objective was to combine two cues, $\mathbf{r}_1$ and $\mathbf{r}_2$, encoding information about the same variable, s, in a statistically optimal way. Assuming a squared error loss function, this can be achieved by computing the mean of the posterior $p(s|\mathbf{r}_1, \mathbf{r}_2)$. For a uniform prior distribution, the posterior mean is given by an expression of the form [23]:

$$\hat{s}_{opt} = \frac{\phi^\top(\mathbf{r}_1 + \mathbf{r}_2)}{\mathbf{1}^\top(\mathbf{r}_1 + \mathbf{r}_2)} \tag{7}$$

where ϕ is the vector of preferred stimuli of input neurons and $\mathbf{1}$ is a vector of ones. During training, the two cues received by the input populations were always non-conflicting: $s_1 = s_2 = s$ and the gains of the input populations varied from trial to trial. The network was trained to minimize the mean squared error between its output and the common s indicated by the two cues.

In the coordinate transformation task, the eye-centered location of an object in 1-d, s_1, was encoded in a population of Poisson neurons with responses $\mathbf{r}_1$ and the current eye position, s_2, was similarly encoded in a population of Poisson neurons with responses $\mathbf{r}_2$. The goal was to compute the head-centered location of the object, which is given by $s = s_1 + s_2$. Assuming uniform priors, the optimal estimate of s can be expressed as [4]:

$$\hat{s}_{\text{opt}} = \frac{\mathbf{r}_1^\top \mathbf{B} \mathbf{r}_2}{\mathbf{r}_1^\top \mathbf{A} \mathbf{r}_2} \tag{8}$$

for suitable matrices $\mathbf{B}$ and $\mathbf{A}$.

In the Kalman filtering task, we considered a one-dimensional time-varying signal evolving according to: $s_t = (1-\gamma)s_{t-1} + \eta_t$, where $\eta_t \sim \mathcal{N}(0, \sigma_\eta^2)$. At each time t, the stimulus was represented by the noisy responses, $\mathbf{r}_{in,t}$, of a population of input neurons with Poisson variability. The input population projected to a recurrent pool of neurons that have to integrate the momentary sensory information coming from the input population with an estimate of the signal at the previous time step (as well as the uncertainty associated with that estimate) to perform optimal estimation of the signal at the current time step. We decoded the estimate of the signal at each time step by a linear read-out of the recurrent pool: $\hat{s}_t = \mathbf{w}^\top \mathbf{r}_{rec,t}$. The network was trained with sequences of length 25 using a squared error loss function. The posterior $p(s_t|\mathbf{r}_{in,1:t})$ is Gaussian with natural parameters given recursively by [4]:

$$\begin{aligned} \frac{\mu_t}{\sigma_t^2} &= \frac{\mu_{in,t}}{\sigma_{in,t}^2} + \frac{(1-\gamma)\mu_{t-1}}{(1-\gamma)^2\sigma_{t-1}^2 + \sigma_\eta^2} \qquad (9) \\ \frac{1}{\sigma_t^2} &= \frac{1}{\sigma_{t,in}^2} + \frac{1}{(1-\gamma)^2\sigma_{t-1}^2 + \sigma_\eta^2} \qquad (10) \end{aligned}$$

where $\mu_{in,t}$ and $\sigma_{in,t}^2$ are the mean and variance of $p(s_t|\mathbf{r}_{in,t})$ which represents the momentary sensory evidence encoded in the input population. These are, in turn, given by $\mu_{in,t} = \phi^\top \mathbf{r}_{in,t}/\mathbf{1}^\top \mathbf{r}_{in,t}$ and $\sigma_{in,t}^2 = \sigma_f^2/\mathbf{1}^\top \mathbf{r}_{in,t}$.

In the binary categorization task, the goal was to classify a noisy orientation measurement into one of two overlapping classes that have the same mean but different variances. Given a noisy activity pattern $\mathbf{r}$ over the input population representing the observed orientation, the posterior probabilities of the two classes can be calculated analytically. The log-likelihood ratio of the two categories is given by [32]:

$$d \equiv \log \frac{p(\mathbf{r}|C=1)}{p(\mathbf{r}|C=2)} = \frac{1}{2}\Big(\log \frac{1+\sigma_2^2 \mathbf{a}^\top \mathbf{r}}{1+\sigma_1^2 \mathbf{a}^\top \mathbf{r}} - \frac{(\sigma_2^2 - \sigma_1^2)(\mathbf{e}^\top \mathbf{r})^2}{(1+\sigma_1^2 \mathbf{a}^\top \mathbf{r})(1+\sigma_2^2 \mathbf{a}^\top \mathbf{r})} \Big) \qquad (11)$$

where $\mathbf{e} = \phi/\sigma_f^2$ and $\mathbf{a} = \mathbf{1}/\sigma_f^2$. The posterior probability of the first class is then given by a sigmoidal function of d: $p(C=1|\mathbf{r}) = 1/(1+\exp(-d))$.

In the causal inference task, the goal was to infer whether two sensory measurements are caused by a common source or by two separate sources. The log-likelihood ratio of these two hypotheses is given by [25]:

$$d = \frac{z_{11}z_{21}}{z_{12}+z_{22}+J_s} - \frac{1}{2}\Big[\frac{z_{22}z_{11}^2}{(z_{12}+J_s)(z_{12}+z_{22}+J_s)} + \frac{z_{12}z_{21}^2}{(z_{22}+J_s)(z_{12}+z_{22}+J_s)} - \log\Big(1 + \frac{z_{12}z_{22}}{J_s(z_{12}+z_{22}+J_s)}\Big)\Big] \quad (12)$$

where J_s is the precision of the Gaussian stimulus distribution and:

$$\begin{aligned} \sigma_f^2 z_{11} &= \phi_1^\top \mathbf{r}_1 \qquad (13) \\ \sigma_f^2 z_{12} &= \mathbf{1}^\top \mathbf{r}_1 \qquad (14) \\ \sigma_f^2 z_{21} &= \phi_2^\top \mathbf{r}_2 \qquad (15) \\ \sigma_f^2 z_{22} &= \mathbf{1}^\top \mathbf{r}_2 \qquad (16) \end{aligned}$$

where σ_f^2 is the common variance of the Gaussian tuning functions of the individual input neurons. ϕ_1 and ϕ_2 are the preferred stimuli of the neurons in the first and second populations respectively. For convenience, we assumed $\phi_1 = \phi_2$. The optimal probability of reporting "same cause" is then simply given as: $p(C=1|\mathbf{r}_1, \mathbf{r}_2) = 1/(1+\exp(-d))$.

In the stimulus demixing task, the goal was to infer the presence or absence of different signal sources in a mixture of signals with unknown concentrations. As a concrete example, the signals can be thought of as different odors, and the task would then be to infer the presence or absence of different odors in an odor mixture with unknown concentrations [4]. Following [4], we assumed a linear mixing model:

$$o_i = \sum_k w_{ik} c_k s_k \qquad (17)$$

where s_k denotes the presence or absence of the k-th odor source, c_k denotes its concentration, o_i is the concentration of the i-th odorant and w_{ik} is the weight of the k-th odor source in the i-th odorant. The task can then be formalized as the computation of the posterior probability of the presence or absence of each odor source, given noisy responses $\mathbf{r} = \{\mathbf{r}_i\}_{i=1}^{n_o}$ of populations of Poisson neurons encoding the odorants: i.e. $p(s_k = 1|\mathbf{r})$. The input populations were assumed to have linear tuning for the odorants: $\mathbf{r}_i \sim \text{Poisson}(o_i \mathbf{f}_i + \mathbf{b}_i)$, where $\mathbf{f}_i$ and $\mathbf{b}_i$ were random vectors with positive entries [4]. As in [4], we assumed four sources and four odorants. The networks were trained to minimize the cross-entropy between the network's outputs and the correct source present/absent labels, s_k.

In the visual search task, the goal was to infer the presence or absence of a target stimulus s_T among a set of heterogeneous distractors. The log-likelihood ratio of the target presence is given by [24]:

$$d = \log \frac{1}{N} \sum_{i=1}^{N} \exp(d_i) \tag{18}$$

where N is the number of stimuli on the display (we assumed $N = 4$) and the local target presence log-likelihoods d_i are given by:

$$d_i = \mathbf{h}_i(s_T)^\top \mathbf{r}_i - \log \Big(\frac{1}{\pi} \int_0^\pi \exp(\mathbf{h}_i(s_i)^\top \mathbf{r}_i) ds_i \Big) \tag{19}$$

For independent Poisson neurons, the stimulus kernel $\mathbf{h}(s)$ is given by $\mathbf{h}(s) = \log \mathbf{f}(s)$, where we assumed von Mises tuning functions for individual input neurons. The integral in the second term on the right hand side was calculated numerically. We trained feed-forward neural networks with two hidden layers on this task (Figure 1h), as networks with a single hidden layer performed poorly.

References

[1] Barron AR (1993) Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans Inf Theory 39(3):930-945.

[2] Battaglia PW, Jacobs RA, Aslin RN (2003) Bayesian integration of visual and auditory signals for spatial localization. JOSA 20(7):1391-1397.

[3] Beck, JM et al. (2008) Probabilistic population codes for Bayesian decision making. Neuron 60:1142-1152.

[4] Beck JM, Latham PE, Pouget A (2011) Marginalization in neural circuits with divisive normalization. J Neurosci 31(43):15310-15319.

[5] Bengio Y, Lee D-H, Bornschein J, Lin Z (2015) Towards biologically plausible deep learning. arXiv:1502.04156.

[6] Britten KH, Shadlen MN, Newsome WT, Movshon JA (1993) Responses of neurons in macaque MT to stochastic motion signals. Vis Neurosci 10:1157-1169.

[7] Caron SJC, Ruta V, Abbott LF, Axel R (2013) Random convergence of afferent olfactory inputs in the Drosophila mushroom body. Nature 497:113-117.

[8] Crochet S, Poulet JFA, Kremer Y, Petersen CCH (2011) Synaptic mechanisms underlying sparse coding of active touch. Neuron 69:1160-1175.

[9] Ernst MO, Banks MS (2002) Humans integrate visual and haptic information in a statistically optimal fashion. Nature 415:429-433.

[10] Fetsch CR, Pouget A, DeAngelis DC, Angelaki DE (2012) Neural correlates of reliability-based cue weighting during multisensory integration. Nat Neurosci 15:146-154.

[11] Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In Proc. 14th International Conference on Artificial Intelligence and Statistics 315-323.

[12] Goldman MS (2009) Memory without feedback in a neural network. Neuron 61(4):621-634.

[13] Graves A, Wayne G, Danihelka I (2014) Neural Turing machines. arXiv:1410.5401.

[14] Haider B, Krause MR, Duque A, Yu Y, Touryan J, Mazer JA, McCormick DA (2010) Synaptic and network mechanisms of sparse and reliable visual cortical activity during nonclassical receptive field stimulation. Neuron 65:107-121.

[15] Haider B, Hausser M, Carandini M (2013) Inhibition dominates sensory responses in the awake cortex. Nature 493:97-100.

[16] Hampshire II JB, Perlmutter BA (1990) Equivalence proofs for multilayer perceptron classifiers and the Bayesian discriminant function. In *Proceedings of the 1990 Connectionist Models Summer School* (Touretzky D, et al. eds.), Morgan Kaufmann.

[17] Heuer HW, Britten KH (2007) Linear responses to stochastic motion signals in area MST. J Neurophysiol 98:1115-1124.

[18] Hillis JM, Watt SJ, Landy MS, Banks MS (2004) Slant from texture and disparity cues: optimal cue combination. J Vis 4(1).

[19] Körding K, Beierholm U, Ma WJ, Quartz S, Tenenbaum JB, Shams L (2007) Causal inference in multisensory perception. PLoS ONE 2(9), e943.

[20] Kwon O-S, Tadin D, Knill DC (2015) A unifying account of visual motion and position perception. Proc Natl Acad Sci USA 112(26):8142-8147.

[21] Leshno M, Lin VY, Pinkus A, Schocken S (1993) Multilayer feed-forward networks with a non-polynomial activation function can approximate any function. Neural Netw 6:861-867.

[22] Lillicrap TP, Cownden D, Tweed DB, Akerman CJ (2014) Random feedback weights support learning in deep neural networks. arXiv:1411.0247.

[23] Ma WJ, Beck JM, Latham PE, Pouget A (2006) Bayesian inference with probabilistic population codes. Nat Neurosci 9(11):1432-1438.

[24] Ma WJ, Navalpakkam V, Beck JM, Berg Rv, Pouget A (2011) Behavior and neural basis of near-optimal visual search. Nat Neurosci 14(6):783-790.

[25] Ma WJ, Rahmati M (2013) Towards a neural implementation of causal inference in cue combination. Multisens Res 26:159-176.

[26] Mante V, Sussillo D, Shenoy KV, Newsome WT (2013) Context-dependent computation by recurrent dynamics in prefrontal cortex. Nature 503, 78-84.

[27] Merfeld DM, Zupan L, Peterka RJ (1999) Humans use internal models to estimate gravity and linear acceleration. Nature 398:615-618.

[28] Morgan ML, DeAngelis GC, Angelaki DE (2008) Multisensory integration in macaque visual cortex depends on cue reliability. Neuron 59:662-673.

[29] Murphy BK, Miller KD (2009) Balanced amplification: a new mechanism of selective amplification of neural activity patters. Neuron 61(4):635-648.

[30] Neal RM (1996) Bayesian learning for neural networks. Lecture Notes in Statistics No. 118, New York: Springer-Verlag.

[31] Parisien C, Anderson CH, Eliasmith C (2008) Solving the problem of negative synaptic weights in cortical models. Neural Comput 20:1473-1494.

[32] Qamar AT, Cotton RJ, George RG, Beck JM, Prezhdo E, et al. (2013) Trial-to-trial, uncertainty-based adjustment of decision boundaries in visual categorization. Proc Natl Acad Sci USA 110(50):20332-37.

[33] Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533-536.

[34] Sclar G, Freeman RD (1982) Orientation selectivity in the cat's striate cortex is invariant with stimulus contrast. Exp Brain Res 46:457-461.

[35] Stettler DD, Axel R (2009) Representations of odor in the piriform cortex. Neuron 63:854-864.

[36] Vinje WE, Gallant JL (2000) Sparse coding and decorrelation in primary visual cortex during natural vision. Science 287:1273-1276.

[37] Williams RJ, Zipser D (1995) Gradient-based learning algorithms for recurrent networks and their computational complexity. In: Chauvin Y, Rumelhart DE (eds.) Back-propagation: Theory, Architectures and Applications. Hillsdale, NJ: Erlbaum.

[38] Wolpert DM, Ghahramani Z, Jordan MI (1995) An internal model for sensorimotor integration. Science 269(5232):1880-1882.

[39] Yamins DLK, Hong H, Cadieu CF, Solomon EA, Seibart D, DiCarlo JJ (2014) Performance-optimized hierarchical models predict neural responses in higher visual cortex. PNAS 111(23):8619-8624.

[40] Zemel R, Dayan P, Pouget A (1998) Probabilistic interpretation of population codes. Neural Comput 10:403-430.

[41] Zipser D, Andersen RA (1988) A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. Nature 331:679-684.

Supplementary Figures

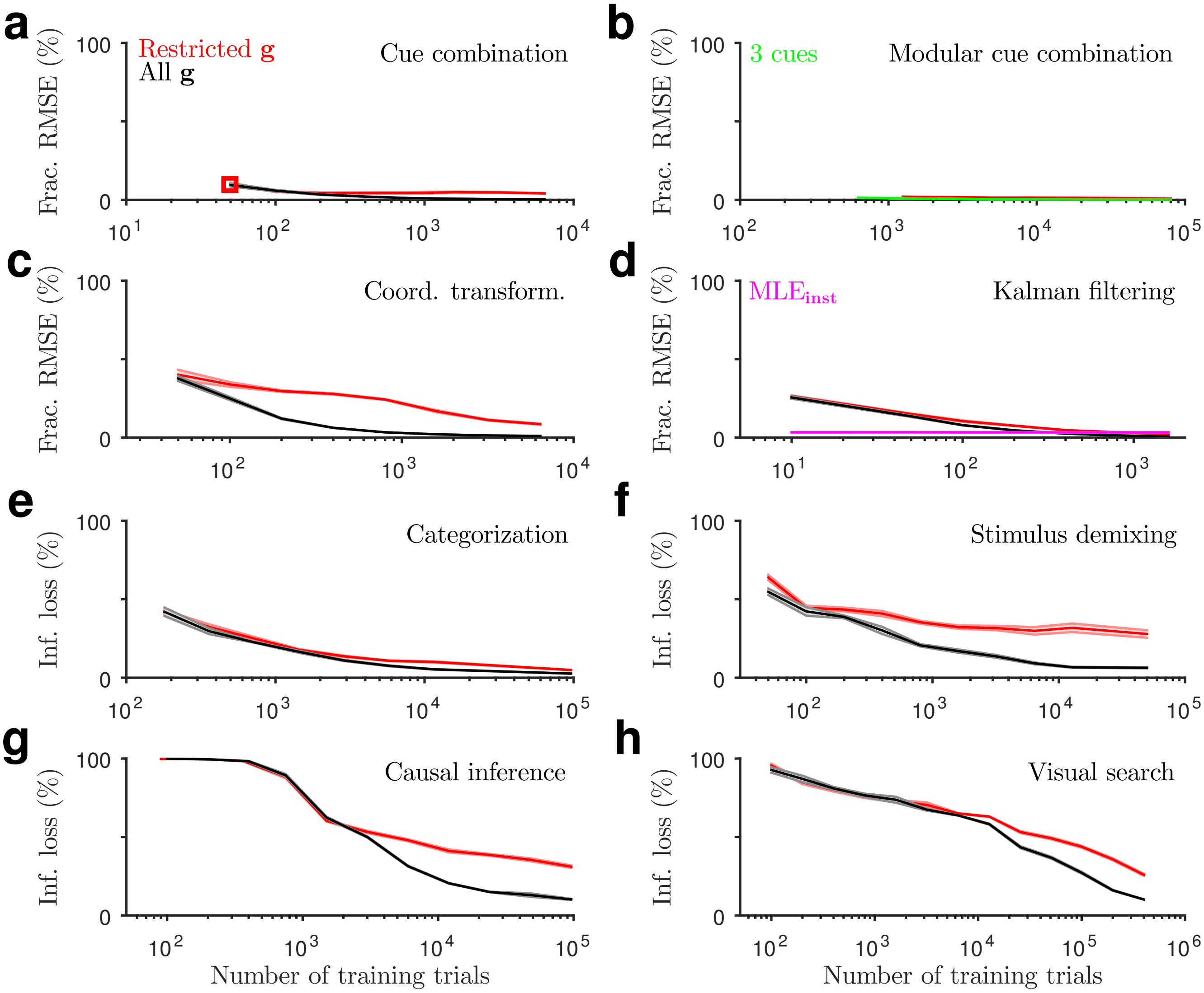

Figure S1: Complete learning curves for the fully trained networks. In (a), the red square indicates the performance of the network whose generalization behavior is shown in Figure 3a. In (b), for comparison with the modular network (Figure 1b), the performance of a standard cue combination network with 3 input populations and a single hidden layer is shown in green. In (d), $\mathrm{MLE}_{\mathrm{inst}}$ refers to a maximum-likelihood estimator that only considers the instantaneous sensory input at each time. The recurrent networks ultimately outperform $\mathrm{MLE}_{\mathrm{inst}}$ in both training conditions.

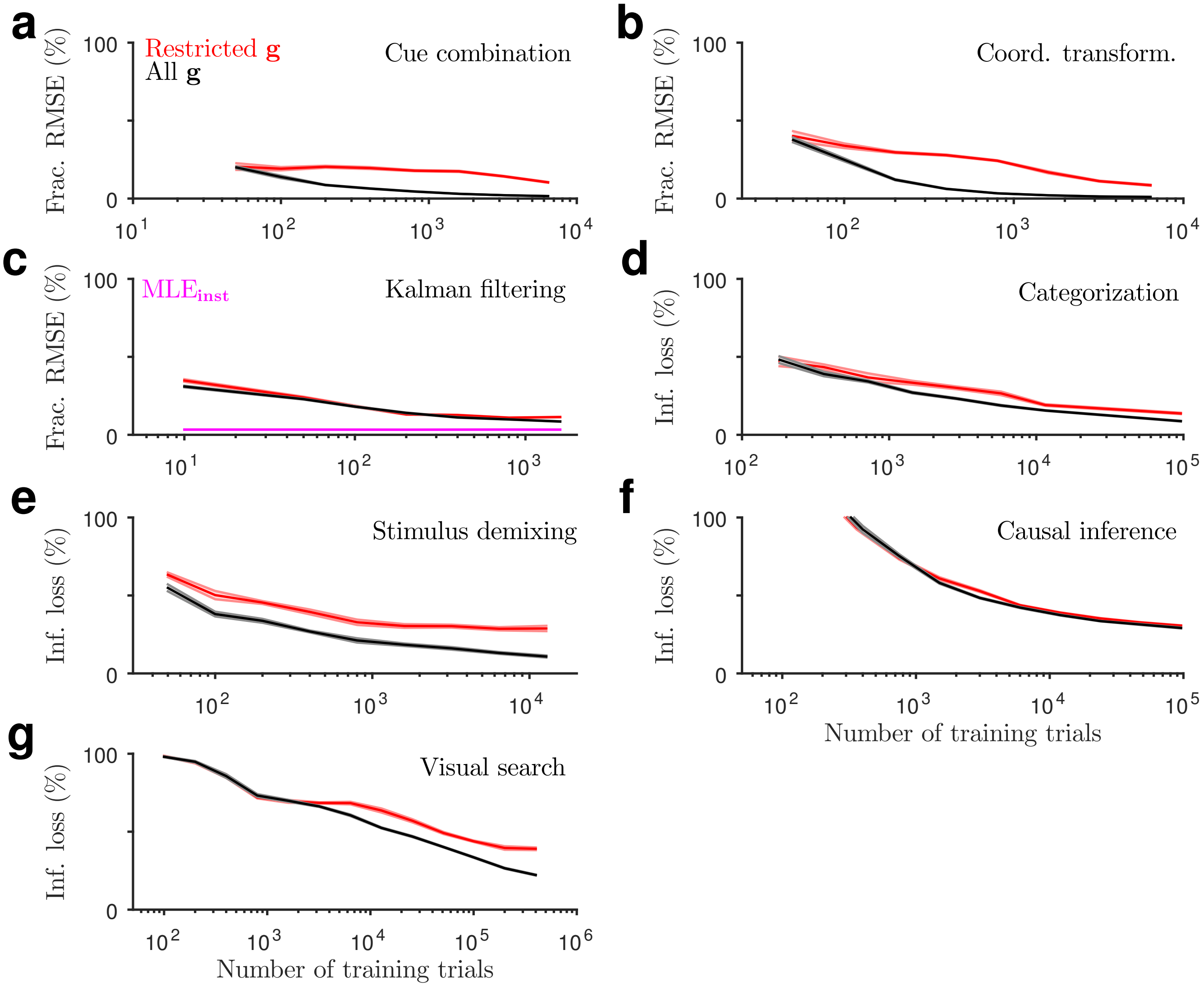

Figure S2: Complete learning curves for the random networks.

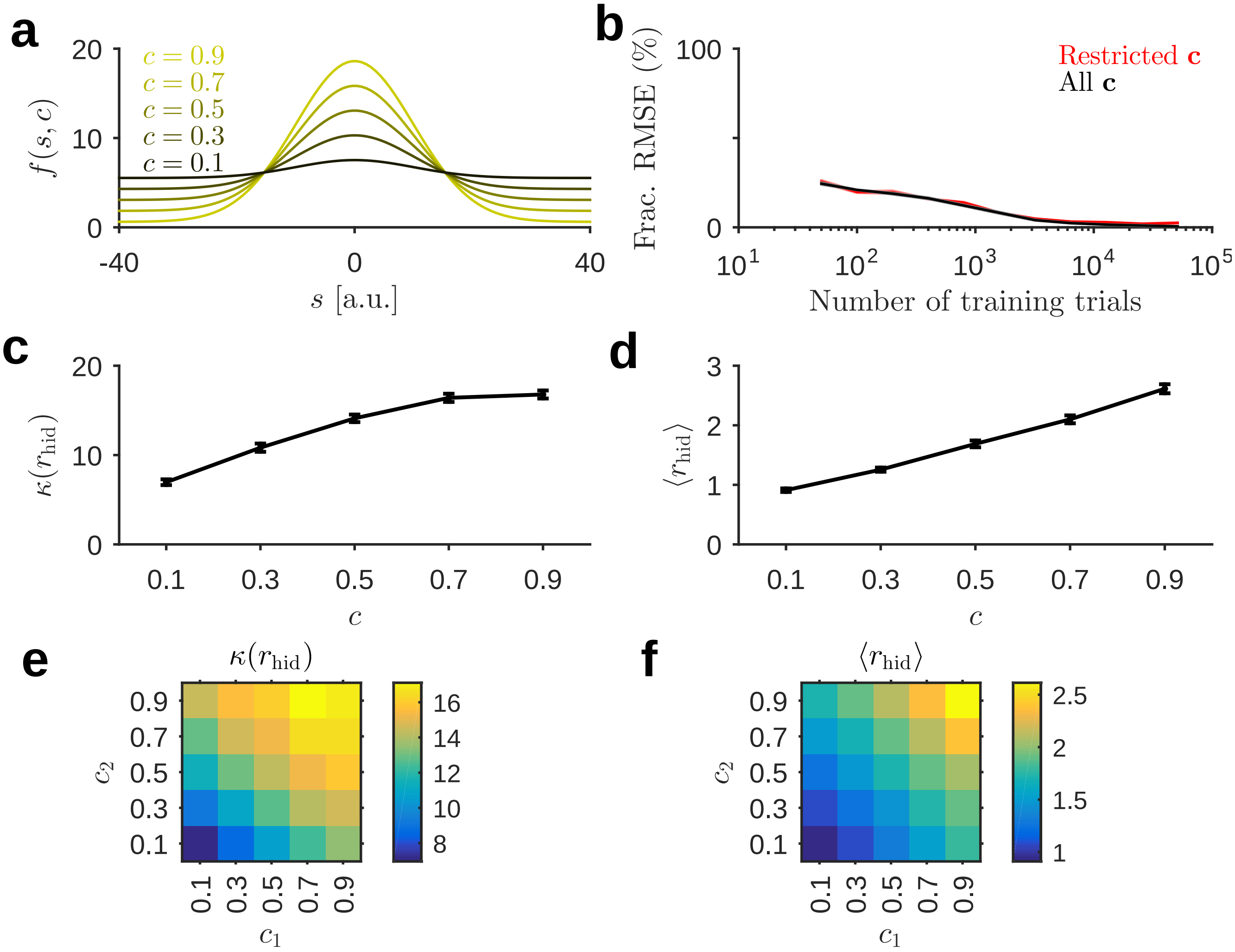

Figure S3: Results for a model with contrast dependent baseline responses as in [10]. The restricted $\mathbf{c}$ training condition corresponds to an equal number of trials of the form $\mathbf{c} = (0.2, 0.2)$ and $\mathbf{c} = (0.8, 0.8)$, whereas the all $\mathbf{c}$ training condition corresponds to all possible combinations of c_1 and c_2 with $c_1, c_2 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ (in equal numbers).

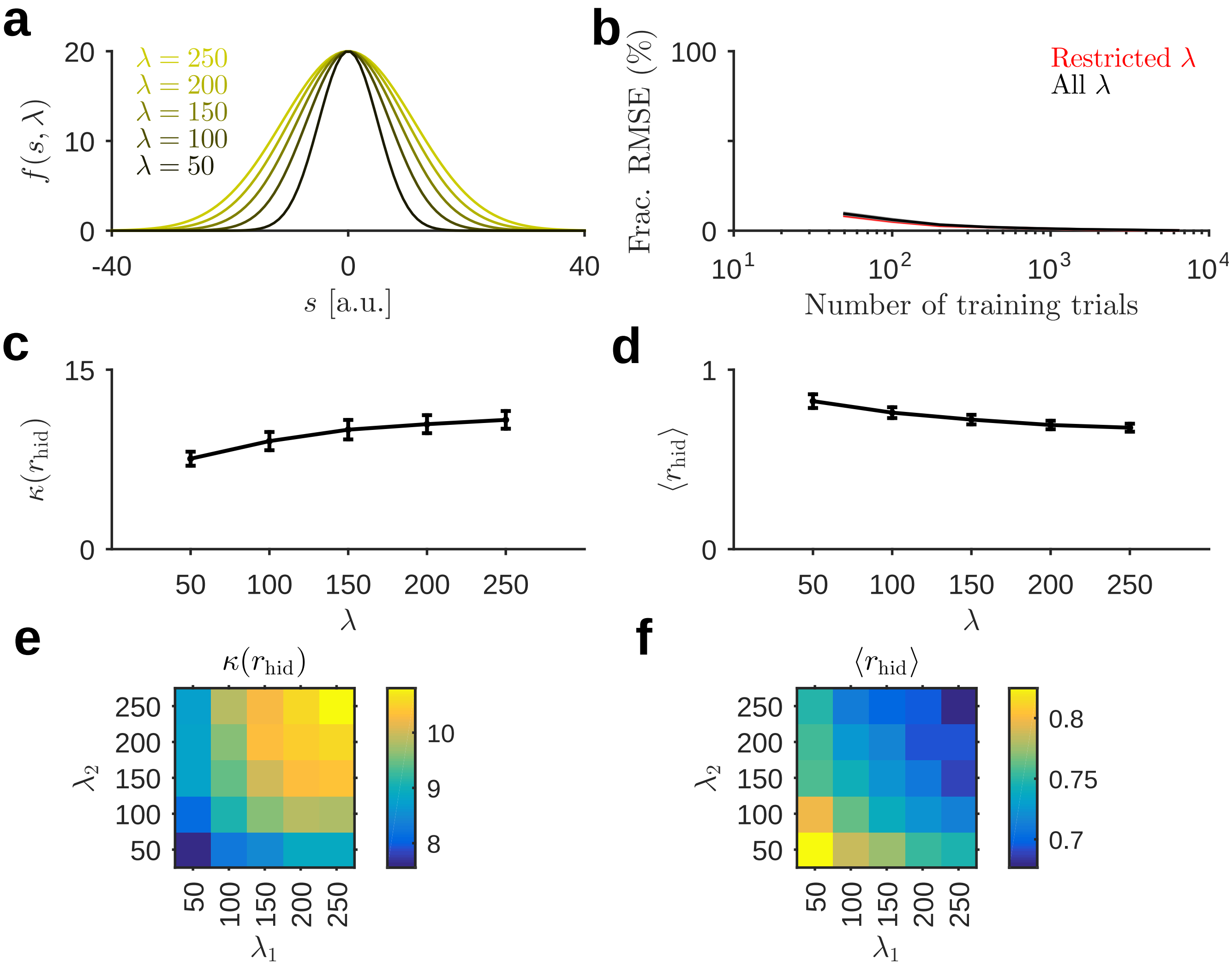

Figure S4: Results for a model where the input tuning widths depend on a nuisance variable λ. Tuning functions were Gaussian: $f(s, \lambda) = \beta \exp(-(s-\phi)^2/\lambda)$. The restricted λ training condition corresponds to an equal number of trials of the form $\lambda = (100, 100)$ and $\lambda = (200, 200)$, whereas the all λ training condition corresponds to all possible combinations of λ_1 and λ_2 with $\lambda_1, \lambda_2 \in \{50, 100, 150, 200, 250\}$ (in equal numbers).

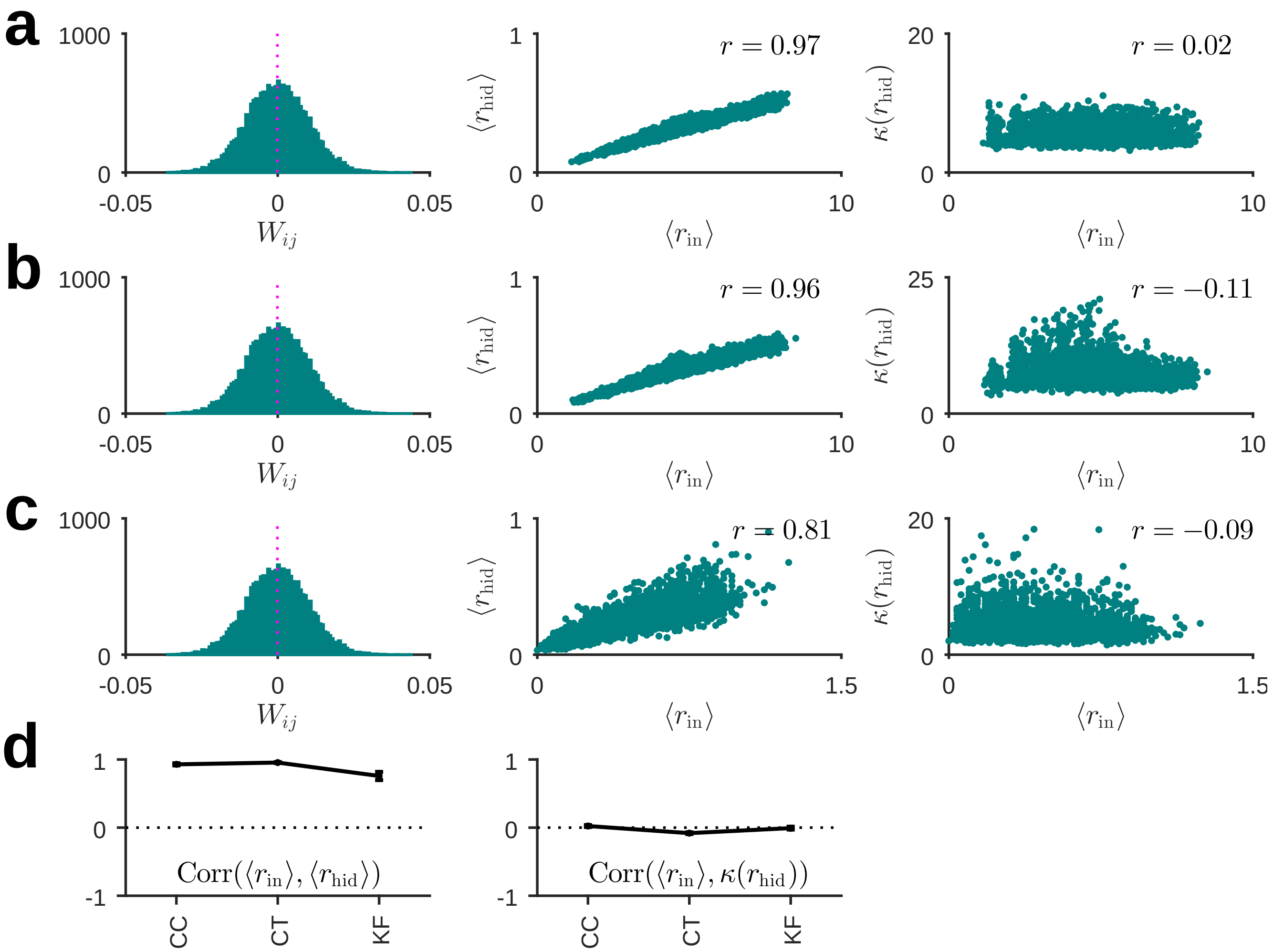

Figure S5: Results for the networks with a normalized linear decoder in the continuous tasks: (a) cue combination, (b) coordinate transformation, (c) Kalman filtering. In these networks, the output of the network was given by $\hat{s} = \mathbf{w}^\top \mathbf{r}_{hid} / \mathbf{1}^\top \mathbf{r}_{hid}$. In (a-c), the first column shows the distribution of input-to-hidden layer weights in the trained networks (the distributions are roughly symmetric around zero), the second column is a scatter plot of the mean input vs. the mean hidden layer activity and the third column is a scatter plot of the mean input vs. the kurtosis of hidden layer activity. (d) Correlation between the mean input and the mean hidden unit activity (left) and the correlation between the mean input and the kurtosis of hidden layer activity in the three tasks. Error bars represent standard errors over 10 independent runs of the simulations.