Performance-optimized neural networks as an explanatory framework for decision confidence

Taylor W. Webb[1,*], Kiyofumi Miyoshi[2], Tsz Yan So[3], Sivananda Rajananda[4], and Hakwan Lau[5,†]

[1]University of California, Los Angeles
[2]Kyoto University
[3]The University of Hong Kong
[4]Harvard University
[5]Center for Brain Science, Riken Institute, Japan
[*]Correspondence to: taylor.w.webb@gmail.com
[†]Correspondence to: hakwan@gmail.com

Abstract

Previous work has sought to understand decision confidence as a prediction of the probability that a decision will be correct, leading to debate over whether these predictions are optimal, and whether they rely on the same decision variable as decisions themselves. This work has generally relied on idealized, low-dimensional modeling frameworks, such as signal detection theory or Bayesian inference, leaving open the question of how decision confidence operates in the domain of high-dimensional, naturalistic stimuli. To address this, we developed a deep neural network model optimized to assess decision confidence directly given high-dimensional inputs such as images. The model naturally accounts for a number of puzzling dissociations between decisions and confidence, suggests a principled explanation of these dissociations in terms of optimization for the statistics of sensory inputs, and makes the surprising prediction that, despite these dissociations, decisions and confidence depend on a common decision variable.

1 Introduction

When faced with a decision, we have the ability not only to choose from a set of possible options, but also to assess how confident we are in our choice. This capacity is an important part of the decision-making process, allowing us to decide whether to gather more information [1], or how much to wager on the outcome of a decision [2]. It has been proposed that this sense of confidence corresponds to an optimal prediction of the probability that a decision will be correct, and that confidence is computed based on the same underlying decision variable as decisions themselves [3–9]. This optimal view of confidence, however, has been called into question by a number of puzzling dissociations between decisions and confidence. These dissociations have led to the formulation of an alternative model in which decisions are made using an optimal 'balance-of-evidence' (BE) rule that incorporates both evidence for and against the decision, whereas confidence is estimated using a simpler heuristic strategy that only considers the 'response-congruent-evidence' (RCE) [10–16]. That is, after weighing the evidence and making a decision, confidence is based only on the evidence in favor of the decision that was made.

These findings raise the question of *why* confidence would be computed using an apparently suboptimal heuristic. This is especially puzzling given findings suggesting that decisions are based on the balance of evidence [14], because it suggests that the evidence against one's choice is available in the decision-making process, but simply not incorporated into confidence judgments. One potential avenue for resolving this puzzle is to reconsider the idealized, low-dimensional settings that have characterized previous approaches to modeling confidence. These approaches require a number of simplifying assumptions that may fail to capture the complex distribution of high-dimensional inputs to the real-world decision-making process, and the question of whether confidence is suboptimal has been shown to depend strongly on these assumptions [17].

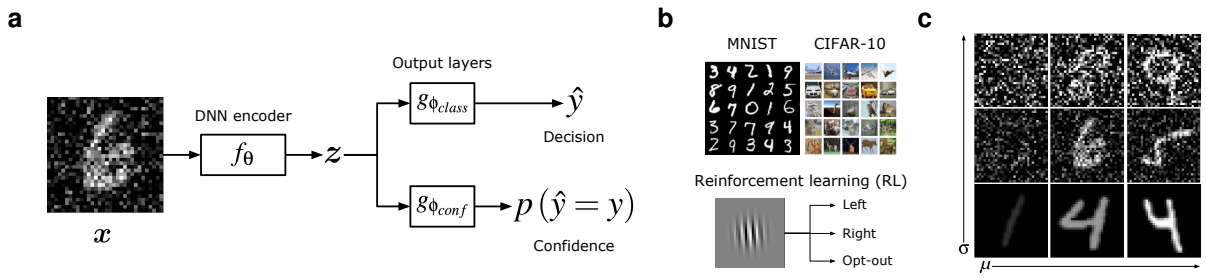

Figure 1: Performance-optimized neural network model of decision confidence. (a) Model architecture. An image $\boldsymbol{x}$, belonging to class y, was passed through a deep neural network (DNN) encoder f_θ to generate a learned low-dimensional representation $\boldsymbol{z}$. This was then passed through two output layers: $g_{\phi_{class}}$ generated a decision $\hat{y}$ classifying the image, and $g_{\phi_{conf}}$ generated a confidence score by predicting $p(\hat{y} = y)$, the probability that the decision was correct. **(b)** The model was trained through supervised learning using the MNIST handwritten digits dataset and the CIFAR-10 object classification dataset. The model was also trained through reinforcement learning (RL) to perform an orientation discrimination task, in which, rather than generating an explicit confidence rating, the model had the option to 'opt-out' of a decision and receive a small but guaranteed reward, allowing the use of the opt-out rate as an implicit measure of confidence. **(c)** To evaluate the relative influence of signal strength and noise on the model's behavior, images were modified by manipulating both the contrast μ and the noise level σ.

In this work, we developed a model of decision confidence that operates directly on naturalistic, high-dimensional inputs, avoiding the need for these simplifying assumptions. To do so, we turn to performance-optimized neural networks, an approach which has recently proven to be a useful framework for both explaining and predicting neural data in a wide range of domains [18–21]. In this approach, rather than attempting to fit model parameters to behavioral or neural phenomena, neural networks models are optimized to perform an ecologically relevant objective, such as recognizing objects or navigating to a reward. To the extent that previously observed phenomena naturally emerge in such models, they may reveal an explanation of those phenomena in terms of optimization for a useful objective. Furthermore, given the complex emergent properties of such models, they can be highly effective tools for generating unexpected novel predictions about neural systems.

We developed a simple neural network model optimized both to make decisions from high-dimensional inputs, and to estimate confidence by predicting the probability those decisions will be correct (Figure 1). Surprisingly, a number of dissociations and seemingly suboptimal features of confidence naturally emerged from the model, including the 'positive evidence' bias, and the condition of 'blindsight' resulting from lesions to primary visual cortex. By manipulating key features of the model's training data, we developed a novel explanation of why these phenomena emerge in such settings, suggesting that they are best viewed not as biases but as adaptive features of optimization for naturalistic stimuli. Finally, by analyzing the computational strategy adopted by the model, we made the surprising discovery that the behavioral and neural dissociations previously taken as evidence in favor of distinct decision variables for decisions and confidence are actually consistent with a common decision variable. These results provide a novel perspective on the computational and neural basis of decision confidence, and suggest new avenues for future investigation.

2 Results

2.1 Accounting for dissociations between decisions and confidence

We first assessed whether the model accounted for a number of previously reported behavioral and neural dissociations between decisions and confidence. Remarkably, despite not being directly optimized to produce these dissociations, the model naturally accounted for them in a manner that closely resembled the previous findings.

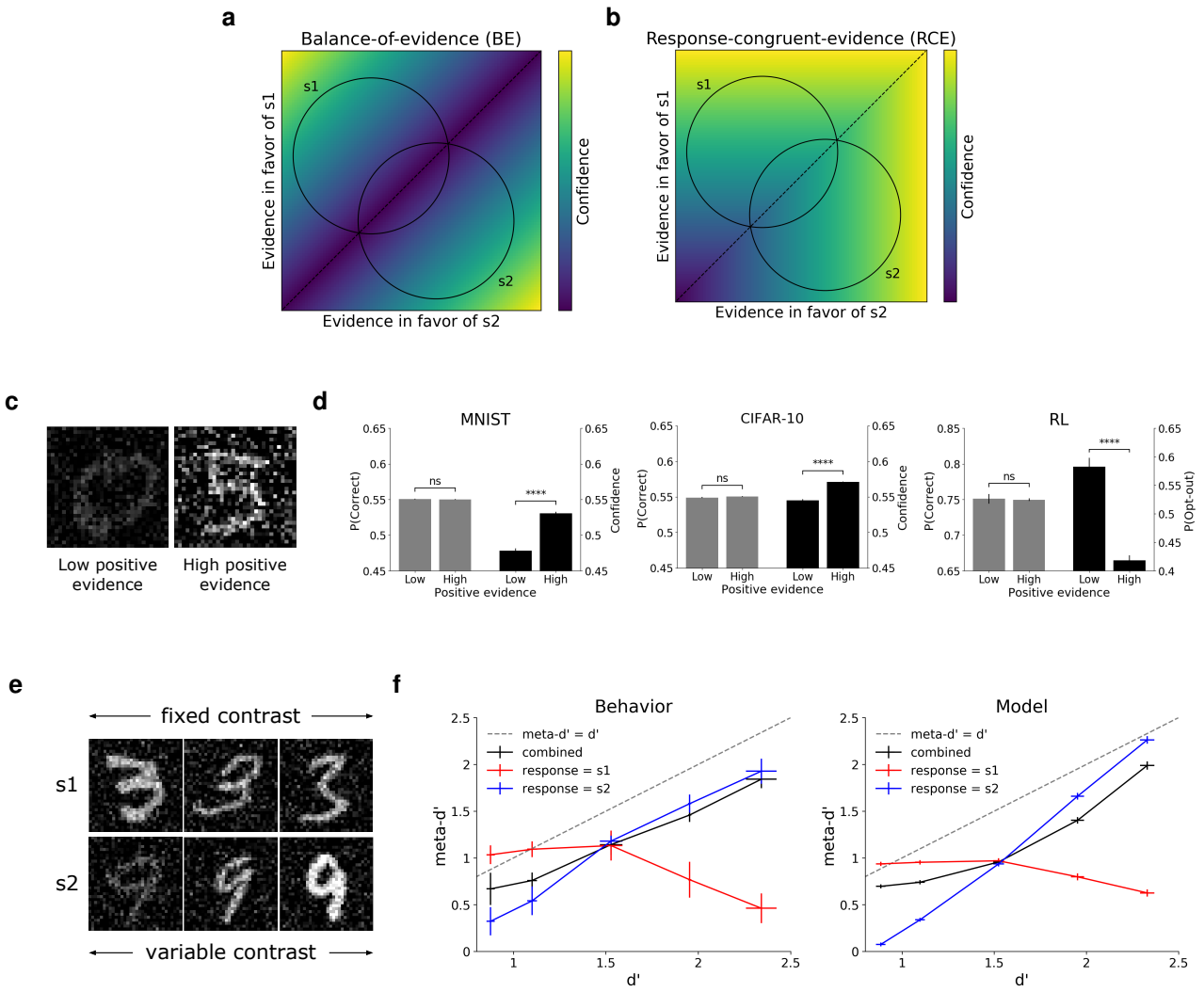

Figure 2: Behavioral dissociations between decisions and confidence. **(a)** Detection-theoretic formalization of confidence in two-choice tasks. Stimuli are modeled as samples from two-dimensional Gaussian distributions, schematized as circles labelled 's1' and 's2', where each dimension represents the evidence in favor of one stimulus category. Given these assumptions, the optimal procedure for estimating confidence is a 'balance-of-evidence' (BE) rule, based on the difference between the evidence in favor of s1 and s2. **(b)** Many results are well modeled by an alternative 'response-congruent-evidence' (RCE) heuristic, according to which, after making a decision, confidence is based entirely on the evidence in favor of the chosen stimulus category, ignoring the evidence in favor of the alternative choice. **(c)** Human and animal decision confidence displays a positive evidence (PE) bias: higher confidence (or lower opt-out rate) in the high vs. low PE conditions despite balanced signal-to-noise ratio and balanced decision accuracy. **(d)** The PE bias naturally emerges in performance-optimized neural networks across multiple datasets, architectures, and learning paradigms. **(e)** Adaptation of behavioral paradigm from Maniscalco et al. [12], s1 is presented at an intermediate contrast, while the contrast of s2 is varied. **(f)** This produces a strong dissociation between type-1 sensitivity (d') and type-2 sensitivity (meta-d'): when participants respond 's1', meta-d' decreases as d' increases (Behavior), a phenomenon which is captured by the neural network model (Model). All neural network results reflect an average over 100 trained networks $\pm$ the standard error of the mean; 'ns' indicates $p > 0.05$, '****' indicates $p < 0.0001$.

2.1.1 The positive evidence bias

Confidence is characterized by a positive evidence (PE) bias [10, 11, 13, 15, 16]. When presented with two conditions, one with low signal and low noise ('low PE' condition, Figure 2c), and the other with high signal and high noise ('high PE' condition), confidence is higher in the high PE condition, despite the fact that signal-to-noise ratio, and therefore decision accuracy, is balanced across these conditions. This bias is considered a key piece of evidence against the optimal BE model (Figure 2a), and in favor of the RCE model (Figure 2b), since an optimal approach should incorporate both the evidence for and against a decision, whereas the RCE model considers only the evidence in favor of a decision.

Figure 2d shows that the PE bias naturally emerged in our model across a range of conditions. For both the MNIST and CIFAR-10 datasets, confidence was higher in the high vs. low PE conditions, despite balanced accuracy, as previously observed in studies of human decision confidence [10, 11, 16]. The presence of this bias therefore did not depend on the specific dataset used, or the architectural details of the model, since experiments on CIFAR-10 used a more complex resnet architecture for the encoder. In the orientation discrimination RL task, the opt-out rate was lower in the high vs. low PE conditions, as previously observed in studies using animal models [13, 15]. The presence of this bias therefore did not depend on the use of supervised learning to train the confidence layer, but also emerged when using a more realistic training signal (reward).

2.1.2 Dissociation between type-1 and type-2 sensitivity

Maniscalco et al. [12] identified and confirmed a more specific, and surprising, prediction of the RCE model: human confidence ratings are, under certain conditions, characterized by a pattern of increasing type-1 sensitivity (as measured by decision accuracy or d') and *decreasing* type-2 sensitivity (as measured by meta-d' [22]). That is, confidence ratings become less diagnostic of decision accuracy as decision accuracy increases. The RCE model predicts that this pattern should emerge whenever a discrimination is made between two stimulus classes, one of which (s1) is presented at a fixed contrast, and one of which (s2) is presented at a variable contrast (Figure 2e). Under these conditions, meta-d' increases as a function of d' for trials on which participants respond 's2', and decreases as a function of d' for trials in which participants respond 's1', resulting in the crossover pattern depicted in Figure 2f ('Behavior'). This pattern is at odds with the BE model, according to which meta-d' should be equal to d' (see Supplementary Figure S1 for further explanation).

We simulated this paradigm in our model using a two-choice variant of the MNIST dataset, in which each model was trained to discriminate between two stimulus classes (e.g. 3 vs. 9). To account for the additional accumulation of noise between the time at which decisions and confidence ratings are made, an additional noise parameter was added to the output of the network's confidence layer. The model showed a strikingly similar pattern to the previously observed results (Figure 2f, 'Model'), capturing both the crossover effect and the pattern of decreasing meta-d' as a function of increasing d' for trials with an 's1' response. Furthermore, even without the addition of the noise parameter, these qualitative effects were still present (Supplementary Figure S2).

2.1.3 Decoding neural evidence for vs. against decisions

In addition to behavioral dissociations, neural data have also indicated that the brain uses distinct decision variables for making decisions and estimating confidence. Peters et al. [14] recorded cortical electrophysiological signals while participants performed a face/house discrimination task, and used machine learning to estimate the amount of neural evidence in favor of faces vs. houses on a trial-by-trial basis. This allowed them to determine whether decisions and confidence were better predicted by the balance of evidence for faces vs. houses, or by the response-congruent evidence alone. Both receiver operating characteristic (ROC; Figure 3a) and choice probability (the area under the ROC curve; Figure 3b) analyses revealed that decisions were better predicted by the BE rule, whereas confidence was predicted about equally well by either the BE or RCE rules, meaning that the incorporation of decision-*incongruent* evidence did not significantly improve the prediction. This pattern suggests that confidence was driven predominantly by the neural evidence in favor of a decision, consistent with the RCE model.

We simulated this analysis in our model by training a separate decoder to predict the stimulus class s1 vs. s2 given the concatenated activation states of all layers in the network. This analysis captured the key qualitative effects present in the original results. Decisions were better predicted by the BE rule than the RCE rule, whereas there was no significant difference between the ability of these rules to predict confidence ratings (Figures 3c and 3d), resulting in a significant interaction between decision rule and dependent variable (decision vs. confidence). We note that,

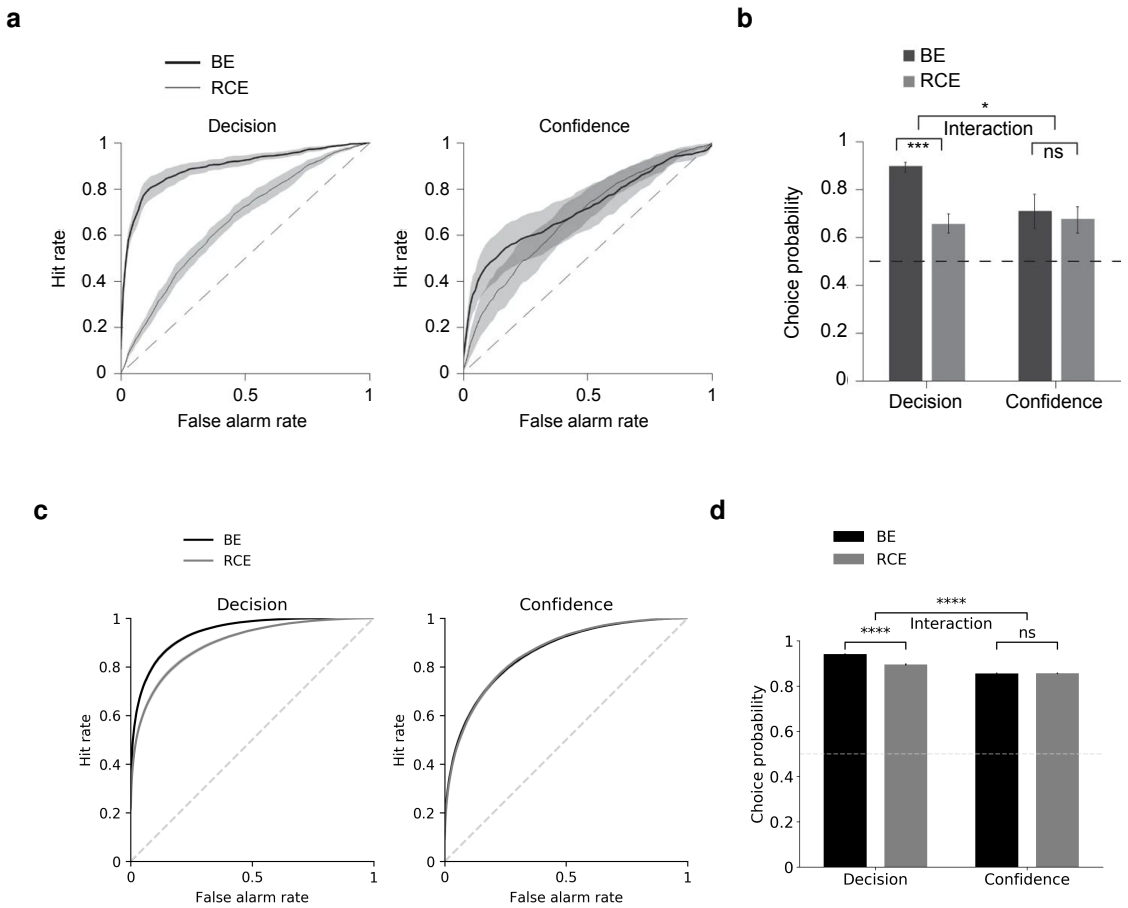

Figure 3: Sensitivity to neural evidence for vs. against decisions. Peters et al. [14] performed a neural decoding analysis to test the predictions of the BE and RCE models. Participants performed a face/house discrimination task, and decoders were trained to use cortical electrophysiological signals to predict the stimulus category (face vs. house). This provided an estimate of the independent neural evidence in favor of faces and houses, which could be used to compute, on a trial-by-trial basis, either the balance of evidence or the response-congruent evidence alone. ROC (panel **(a)**) and choice probability (area under the ROC curve; panel **(b)**) analyses were performed to determine to what extent these neural evidence measures could predict decisions and confidence ratings. Decisions were predicted significantly better by the BE rule than the RCE rule. Confidence ratings were predicted equally well by both rules. This same pattern was displayed by the neural network model (panels **(c)** and **(d)**). Neural network results reflect an average over 100 trained networks $\pm$ the standard error of the mean; 'ns' indicates $p > 0.05$, '*' indicates $p < 0.05$, '***' indicates $p < 0.001$, '****' indicates $p < 0.0001$.

although these qualitative effects were captured by the model, the quantitative difference between the ability of the BE and RCE rules to predict decisions was not as large as in the original results, which may suggest a potential discrepancy between the brain and our model. However, another potential source of the discrepancy may be the different sources of signal and noise present in our simulated decoding analyses vs. real decoding analyses using electrophysiological data with relatively low spatial resolution.

2.1.4 Blindsight

One particularly striking dissociation between decisions and confidence comes from the condition known as 'blindsight' [23], in which a lesion to primary visual cortex (V1) results in the subjective sensation of blindness despite relatively preserved visual capacity. One way to characterize blindsight is as a severe deficit in visual metacognition: blindsight patients have very low confidence in their visual discriminations, and their confidence ratings are generally

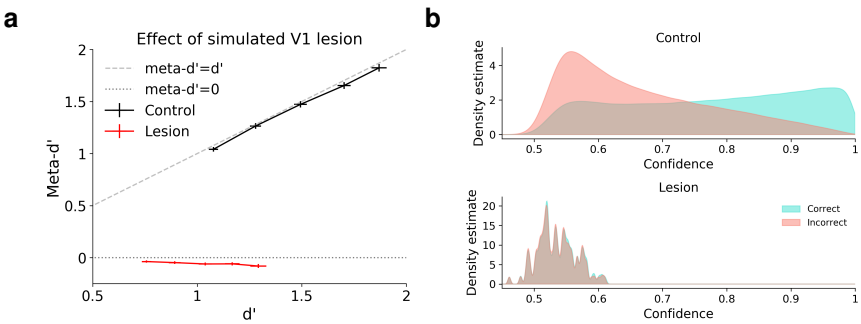

Figure 4: Simulating 'blindsight'. Lesions to V1 can cause the condition known as 'blindsight', in which patients have the subjective sensation of blindness despite preserved visual capacity. This pattern can be formalized as a combination of preserved type-1 sensitivity (d'), low visual confidence, and low type-2 sensitivity (meta-d'). **(a)** Lesions to V1 were simulated in the model by scaling activations in the first layer of the trained network by a factor of 0.01. This resulted in a sharp reduction in meta-d' despite relatively preserved d'. **(b)** Confidence ratings were significantly lower following simulated lesions, and the distribution of confidence ratings showed a nearly complete overlap for correct and incorrect trials, consistent with meta-d' ≈ 0. All results reflect an average over 100 trained networks $\pm$ the standard error of the mean.

not very predictive of whether those discriminations will be correct or incorrect [24, 25]. This can be formalized as a pattern of relatively preserved d' combined with very low meta-d'. This pattern cannot be explained by the optimal (BE) approach to decision confidence, according to which meta-d' should be equal to d'.

We simulated lesions to V1 by scaling the activations in the first layer of the trained neural network model by a factor of 0.01. The small amount of signal remaining in the first layer of the network was intended to model the intact visual signals from subcortical regions that are thought to mediate residual visual function in blindsight [26]. We found that, despite this significant scaling of activations, the model retained substantial visual function, as indicated by high d' values, whereas the model's confidence ratings were no longer predictive of performance, as indicated by meta-d' ≈ 0 (Figure 4a). This is in contrast to control networks, without a lesion, which displayed meta-d' $\approx$ d'. We note that, in our analysis, confidence criteria were selected based on the empirical distribution of confidence outputs for each condition, so meta-d' ≈ 0 cannot be trivially explained by confidence ratings that all fall below an arbitrary pre-lesion criterion. We also computed density estimates for confidence ratings on correct vs. incorrect trials, which further confirmed that, in lesioned networks, confidence ratings were not only low but also had completely overlapping distributions for – and thus could not discriminate between – correct and incorrect trials (Figure 4b). The model therefore captured the key metacognitive components of blindsight.

2.2 Apparent biases reflect optimization for naturalistic settings

The previous results show that a number of specific signatures of decision confidence naturally emerge from performance-optimized neural networks, despite the fact that these signatures have previously been viewed as evidence of suboptimal processes. This raises two possibilities. One is that these signatures emerge in such networks despite their suboptimality, perhaps due to some architectural limitation or flaw in the optimization process. An alternative possibility is that the observed dissociations and biases are actually adaptive consequences of optimization for the statistics of naturalistic stimuli, which violate the assumptions of standard models in ways that have not been previously appreciated. Here, we present evidence in favor of this latter interpretation.

To adjudicate between these two possibilities, we focused on the PE bias as a case study. We investigated four regimes, distinguished by the distribution of contrast and noise levels in the training data. Networks trained on the standard regime, involving variation in both contrast and noise levels, displayed a PE bias (Figure 5a, 'Standard training'). Networks trained on data with a fixed contrast-to-noise ratio ('Fixed μ/σ training'), equivalent to being trained directly on the low and high PE conditions in which decision accuracy is balanced, did not display any bias. When networks were trained on a regime in which contrast was fixed and only noise varied ('Fixed μ training'), a *reversed* PE bias was observed. That is, confidence was higher in the low vs. high PE conditions. Under this training regime, accuracy is primarily a function of the noise level, so it makes sense to adopt a confidence strategy

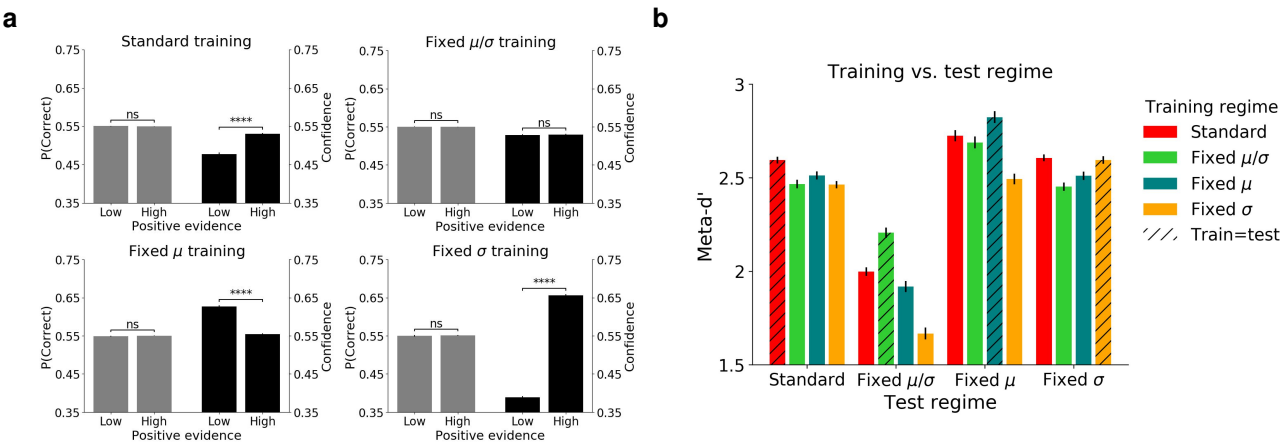

Figure 5: The PE bias is driven by distribution of training data. (a) Effect of four different training regimes on presence of PE bias. Variance in both contrast and noise (standard training) results in PE bias. Fixed contrast-to-noise ratio (fixed μ/σ training) results in no PE bias. Variance in noise only (fixed μ training) results in reversed PE bias – confidence is higher in low vs. high PE conditions. Variance in contrast only (fixed σ training) results in significantly larger PE bias than standard training regime. Results reflect an average over 100 trained networks $\pm$ the standard error of the mean; 'ns' indicates $p > 0.05$, '****' indicates $p < 0.0001$. **(b)** Metacognitive performance (as measured by meta-d') in each test regime as a function of a network's training regime. In general, for each test regime, the best metacognitive performance was achieved by networks trained in that regime (as indicated by hatched bars; note that distinct training and test images were used even when the training and test regime were the same). The one exception is the fixed σ test regime, on which networks trained in both the standard and fixed σ regime – both of which displayed a PE bias – performed equally well. Results reflect an average over 200 trained networks $\pm$ the standard error of the mean; statistical comparisons are omitted for visual clarity, but are presented in Supplementary Table S1.

based primarily on the level of sensory noise, resulting in higher confidence in the low PE (low noise) condition. By contrast, under a regime in which noise is fixed and only contrast varies ('Fixed σ training'), it makes sense to adopt a confidence strategy based primarily on the contrast level. Consistent with this, networks trained on this regime showed a PE bias approximately five times as large as in the standard training regime (average confidence difference between the low and high PE conditions of 0.27 ± 0.005 in the fixed σ regime vs. 0.05 ± 0.004 in the standard regime).

These results reveal how the statistics of the training data, and in particular variability in signal strength, can have a dramatic effect on the presence or absence of a specific metacognitive bias. Furthermore, the confidence strategies adopted in each training regime generally resulted in superior metacognitive performance for that regime (Figure 5b). Thus, these results suggest that such biases may reflect a metacognitive strategy that is rational within the context of a particular statistical regime. This in turn raises the intriguing possibility that these biases are present in human decision confidence as a consequence of optimization for the statistics of sensory inputs, which are characterized by variable signal strength [27].

2.3 Learned confidence strategy not fully explained by simple decision rules

Next we sought to understand whether the model's behavior was explainable by simple, low-dimensional decision rules, such as RCE. To answer this question, we characterized the complete pattern of confidence as a function of the underlying stimulus evidence, taking inspiration from a paradigm developed by Koizumi et al [11]. In that study, participants were tasked with determining which of two superimposed stimuli (left and right tilted gabor patches) had a higher contrast. This allowed for the independent manipulation of the evidence in favor of each stimulus class, permitting them to test how confidence behaves under conditions which may not naturally arise, such as a scenario in which there is strong evidence in favor of both stimulus classes. Here, we took this paradigm a step further, exploiting the ability to test neural networks on a very large number of trials, and evaluated confidence over the full space of evidence in favor of each stimulus class.

After training the model on the two-choice variant of MNIST, we presented it with images consisting of overlapping

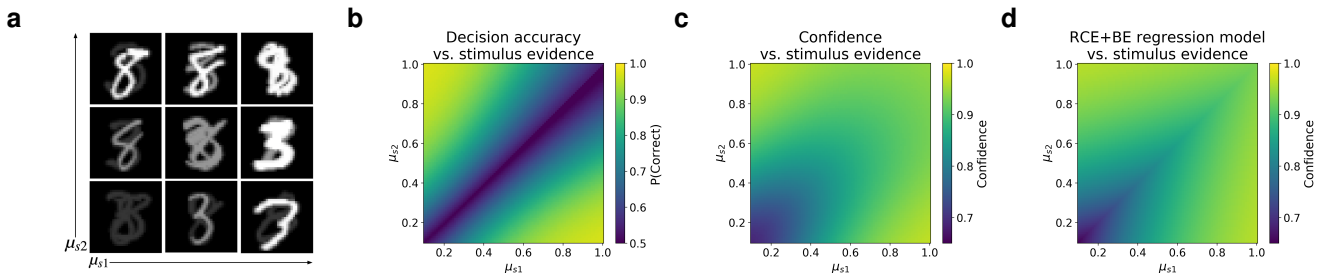

Figure 6: Analysis of learned decision rules. (a) Stimuli used to estimate accuracy and confidence as a function of stimulus evidence. Each image consists of two superimposed digits belonging to classes s1 and s2, with independently varying contrast levels (μ_{s1} and μ_{s2}). **(b)** Decision accuracy resembled the BE rule. **(c)** Confidence displayed a more complex pattern. **(d)** Confidence was best predicted by a multiple regression model incorporating both the RCE and BE rules. Notably, even this combination of decision rules could not entirely explain the network's confidence strategy, as can be seen by comparing panels **(c)** and **(d)**, indicating that the network learned a more complicated strategy that could not be fully described by simple decision rules. Results reflect an average over 100 trained networks.

digits (Figure 6a), independently varying the contrast of each digit (μ_{s1} and μ_{s2}). We then computed both decision accuracy and confidence over this space. Decision accuracy resembled the BE rule (Figure 6b), whereas confidence displayed a more complex pattern (Figure 6c). To better understand the model's confidence behavior, we formally compared the pattern displayed in Figure 6c to regression models based on the BE and RCE rules. Confidence was better explained by the RCE rule than the BE rule (RCE $R^2 = 0.76 \pm 0.01$; BE $R^2 = 0.42 \pm 0.01$; paired t-test, RCE vs. BE, $t = 17.98$, $p < 0.0001$), but a multiple regression incorporating both rules explained the model's behavior better than either rule alone (RCE+BE $R^2 = 0.84 \pm 0.01$, paired t-test, RCE+BE vs. RCE, $t = 9.49$, $p < 0.0001$). Furthermore, even the combination of these rules did not fully capture the pattern of confidence ratings generated by the neural network, as can be seen by comparing the patterns in Figures 6c and 6d. This indicates that the neural network model learned a confidence strategy that was more complex than these simple decision rules.

2.4 Dissociations are consistent with common decision variable

In order to better understand the computational mechanisms adopted by the model, we performed principal component analysis on the representations in the penultimate layer ('$\boldsymbol{z}$' in Figure 1a) of networks trained on a two-choice task. We found that, despite the fact that the model captured a wide range of dissociations between decisions and confidence, the model's learned representations could be characterized by a two-dimensional geometry involving only a single decision variable. The top two principal components accounted for more than 97% of the variance (Figure 7a). Principal component 1 (PC1) predicted both the stimulus class s1 vs. s2 (Figure 7b; prediction accuracy $= 0.88 \pm 0.003$), and the network's decision output (Figure 7e; $R^2 = 0.99 \pm 0.001$). Principal component 2 (PC2) predicted both whether a decision was correct vs. incorrect (Figure 7c; prediction accuracy $= 0.88 \pm 0.003$), and the network's confidence output (Figure 7f; $R^2 = 0.9 \pm 0.01$). Most surprisingly, PC2 corresponded closely to a rectification of PC1 (i.e. PC2 $\propto$ |PC1|; Figure 7d; $R^2 = 0.86 \pm 0.014$), suggesting that these components represented essentially the same decision variable.

How can this representational scheme be reconciled with the observed dissociations between decisions and confidence? One possibility is that PC1, which forms the basis of the network's decisions, is subject to the same biases as confidence, albeit in a manner that is not detectable in the decisions themselves. We found strong evidence for this interpretation. Specifically, we found that the PE bias was displayed not only by PC2 (Figure 8b), as would be expected by the bias in the model's confidence outputs (Figure 8a), the bias was also present in both the rectification of PC1 (|PC1|), as well as a rectified version of the network's continuous decision output capturing the probability assigned by the network to the chosen decision (i.e. the distance of the decision output from 0.5). Thus, this bias was manifested in the magnitude of the probability assigned to the chosen decision, but did not affect the decision itself, which is based only on the most likely choice.

Furthermore, although the means of the distributions for classes s1 and s2 had greater separation along PC1 in the high vs. low PE conditions (which follows from the fact that |PC1| is subject to the PE bias), these distributions

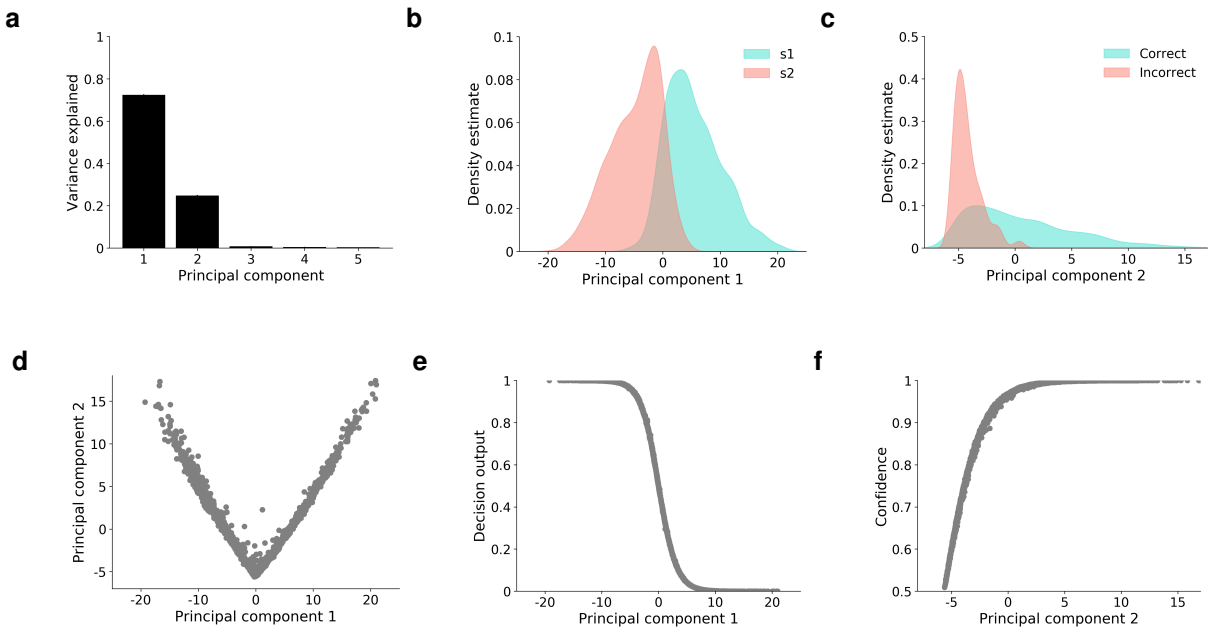

Figure 7: Learned representations extract a common variable for both decisions and confidence. (a) Principal component analysis revealed that the learned representations in the model's penultimate layer were almost entirely explained by the top two principal components. Panel **(a)** reflects an average across 100 trained networks ± the standard error of the mean. Panels **(b)-(f)** depict results from a single trained network for the purposes of illustrating the model's learned two-dimensional geometry, but this general representational scheme was shared by all networks. **(b)** Kernel density estimates show that the distributions for s1 vs. s2 were were separated along PC1. **(c)** The distributions for correct vs. incorrect trials were separated along PC2. **(d)** PC2 closely resembled a rectification of PC1. **(e)** PC1 predicted the model's decisions. **(f)** PC2 predicted the model's confidence ratings. Note that panels **(b)-(f)** depict trial-by-trial data points, not the predictions of a regression model.

also had greater variance in the high PE condition (Figures 8c). This explains how it is possible for decision accuracy to be balanced in these conditions, despite the use of a common decision variable subject to the PE bias – accuracy is determined both by the signal strength of the decision variable, which is higher in the high PE condition, *and* the noise level which is *also* higher in the high PE condition. This explanation mirrors previously proposed accounts according to which dissociations between confidence and decision accuracy emerge due to the use of fixed criterion applied to a unidimensional decision variable [28, 29].

It is important to emphasize however that, unlike these previous accounts, the common decision variable extracted by our model does not represent a simple balance of sensory evidence in favor of each stimulus class. To illustrate this, we applied the analysis described in Section 2.3, and found that |PC1| (Figure 8d), on which decisions were based, displayed a nearly identical pattern to PC2 (Figure 8e), on which confidence was based. Furthermore, the rectified decision output (Figure 8f) displayed a nearly identical pattern to the confidence outputs themselves (Figure 8g)[1]. Most importantly, these results show that the common decision variable extracted by the network represents a complex function of the two-dimensional sensory evidence space, rather than a simple balance-of-evidence. As with the network's confidence output, this complex function bares some resemblance to the RCE rule, explaining how it is possible for the network to exhibit RCE-like behavior despite the use of a common decision variable. These analyses lead to a surprising prediction of the model: the decision variable underlying decisions (which could be recovered from neural data by e.g. training a decoder to predict decisions) should be subject to the same biases as confidence, despite the fact that those biases do not characterize decisions themselves.

Finally, we note that the two-dimensional geometry described here is at odds with the decoding results presented in Section 2.1.3. This can be explained by the fact that the decoders in that analysis were trained using inputs

[1] A sigmoid nonlinearity is applied in between the penultimate layer and the output layers (see Methods Section 4.3.2). Thus, the patterns displayed by the rectified decision output and confidence reflect nonlinear transformations of |PC1| and PC2 respectively.

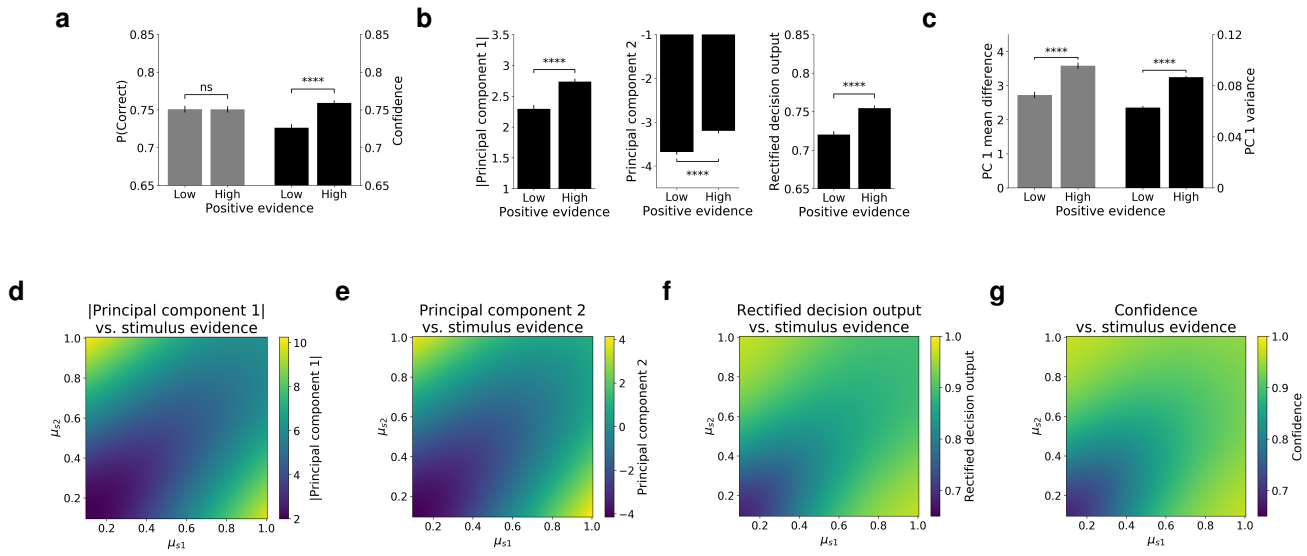

Figure 8: Common decision variable is subject to same biases as confidence. **(a)** Confidence is subject to the PE bias in a two-choice task, despite balanced decision accuracy. **(b)** The common decision variable, used for both decisions and confidence, is subject to the PE bias, as seen in the rectified values along PC1 (|PC1|), PC2, and the rectified values of the network's decision output (the probability that the network assigns to the chosen decision), but cannot be seen in decisions themselves, which only take into account the most likely choice. **(c)** The means of the distributions for s1 and s2 along PC1 are more separated in the high vs. low PE conditions, but these distributions also have greater variance in the high PE condition, explaining the balanced accuracy across these conditions. Panels **(a)**-**(c)** reflect an average over 100 trained networks ± the standard error of the mean. **(d)**-**(g)** Visualizing the decision variable as a function of the evidence in favor of each stimulus class. **(d)** |PC1| displays a nearly identical pattern to **(e)** PC2. **(f)** Rectified decision output displays a nearly identical pattern to **(g)** confidence. Note that the rectified decision output and confidence differ from |PC1| and PC2 due to the addition of a sigmoid nonlinearity (see Methods Section 4.3.2). Panels **(d)**-**(g)** reflect an average over 100 trained networks.

from all layers of the network, mirroring the whole-brain decoding approach in the original study [14], whereas the principal component analysis described here was applied only to the penultimate layer of the network. We applied that decoding analysis to the penultimate layer, and found that both decisions and confidence were better predicted by the balance of neural evidence than by the response-congruent evidence alone (Supplementary Figure S3), consistent with the presence of a common decision variable that implements a complex combination of both the BE and RCE rules. Thus, one possible resolution of this discrepancy is that these previous results are driven primarily by earlier sensory regions, from which neural signals are generally more easily decoded than in higher-order decision-making regions [30]. This suggests the need for direct intracellular recordings from decision-making areas, e.g. lateral intraparietal cortex (LIP) or dorsolateral prefrontal cortex (dlPFC), to test the predictions of the model.

3 Discussion

The question of whether confidence judgments reflect an optimal prediction of the probability of being correct has been hotly debated in recent years [3–17, 31, 32]. Those debates have generally centered around models that operate over idealized, low-dimensional inputs. In this work, by contrast, we used performance-optimized neural networks to study confidence in the context of realistic, high-dimensional stimuli. We were surprised to find that a relatively simple model, optimized only to predict its own likelihood of being correct, captured many of the biases and dissociations that have driven recent debates.

Our findings have an important conceptual link to the work of Miyoshi & Lau [17], who found that the RCE rule is advantageous when stimulus distributions are characterized by greater target variance than non-target variance (additional variance in the evidence for the correct stimulus class). In our work, we employed a training regime that

consisted of images with varying contrast, which introduces an additional degree of variance in the target dimension not present in the non-target dimension. Consistent with the conclusions of Miyoshi & Lau, we found that the PE bias could be eliminated, or even reversed, by manipulating this component of the model's training data. These results underscore the importance of closely examining the assumptions on which previous theories have been built, and suggest that seemingly suboptimal features of decision confidence may in fact have a rational basis.

Our analyses revealed that the model adopted a complex nonlinear strategy for rating confidence. Given that this strategy was not entirely explainable by simple decision rules, what is the best way to understand this pattern? We suggest that this pattern is best understood as a data-driven approximation of the model's training objective – to predict $p(\hat{y} = y|\boldsymbol{x})$, the probability of being correct given the sensory input. This data-driven approximation will tend to diverge from an exact analytical solution to this objective (which is intractable in such high-dimensional spaces) to the extent that the model is evaluated on conditions outside those it was trained on. This may explain, for instance, why the model displayed high confidence in conditions under which there is strong evidence for two alternative stimulus classes (the upper right quadrant in Figure 6c), despite very low decision accuracy under these conditions. This scenario is highly improbable under the model's training regime, and so the model's behavior is effectively an extrapolation. Along similar lines, this may explain why the model displayed such low confidence in the simulated 'blindsight' condition, despite relatively preserved decision accuracy, since, in the model's training regime, conditions under which there is only very weak evidence in favor of either stimulus class typically result in low accuracy. More work is needed to develop a principled understanding of how neural networks generalize in these settings [33]. Future work should also investigate whether human confidence ratings are best explained by simple decision rules or by a more flexible pattern akin to the behavior of our model.

The representational scheme learned by the model, in which both confidence and decisions are represented using a common decision variable, bares a striking resemblance to previous results in decision-making regions such as LIP and the superior colliculus [3, 15, 34]. In contrast to this common decision variable account, it has recently been proposed that confidence and decisions are supported by distinct populations of neurons with differing degrees of lateral inhibition, accounting for the observed dissociations between decisions and confidence [35]. A key contribution of our model is to show how these dissociations can arise despite the use of a common decision variable. This is due in part to the application of a fixed confidence criterion to this decision variable, consistent with some previous models [28, 29]. However, a more surprising feature of our model is that this learned decision variable represents the underlying sensory evidence space in a complex nonlinear manner, partially resembling the RCE rule, rather than reflecting a simple balance of the sensory evidence. These considerations lead to the novel prediction that the neural decision variable underlying decisions should be subject to the same biases as confidence.

We have presented a simple, high-level model that abstracts over many important properties of biological neural networks, including the presence of distinct cell types, temporal dynamics, etc. Though this level of abstraction has proven useful, it will also be important to expand the functionality and biological detail of the model in future work. For instance, the model could be expanded by replacing the feedforward encoder with a recurrent network [36], allowing the model to make contact with the rich body of data on temporal evidence accumulation in decision-making [37, 38]. Furthermore, recently developed techniques [39] could be used to incorporate distinct inhibitory and excitatory cell types, allowing the model to implement lateral inhibition, which is thought to play a key role in the evidence accumulation process [40], and also figures prominently in mechanistic accounts of decision confidence [35]. The present approach therefore can be viewed as a general explanatory framework that suggests a number of exciting prospects for future work.

References

[1] T. Balsdon, V. Wyart, and P. Mamassian, "Confidence controls perceptual evidence accumulation", *Nature communications*, vol. 11, no. 1, pp. 1–11, 2020.

[2] N. Persaud, P. McLeod, and A. Cowey, "Post-decision wagering objectively measures awareness", *Nature neuroscience*, vol. 10, no. 2, pp. 257–261, 2007.

[3] R. Kiani and M. N. Shadlen, "Representation of confidence associated with a decision by neurons in the parietal cortex", *Science*, vol. 324, no. 5928, pp. 759–764, 2009.

[4] A. Kepecs and Z. F. Mainen, "A computational framework for the study of confidence in humans and animals", *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 367, no. 1594, pp. 1322–1337, 2012.

[5] C. R. Fetsch, R. Kiani, W. T. Newsome, and M. N. Shadlen, "Effects of cortical microstimulation on confidence in a perceptual decision", *Neuron*, vol. 83, no. 4, pp. 797–804, 2014.

[6] F. Meyniel, D. Schlunegger, and S. Dehaene, "The sense of confidence during probabilistic learning: A normative account", *PLoS computational biology*, vol. 11, no. 6, e1004305, 2015.

[7] S. Gherman and M. G. Philiastides, "Neural representations of confidence emerge from the process of decision formation during perceptual choices", *Neuroimage*, vol. 106, pp. 134–143, 2015.

[8] A. Pouget, J. Drugowitsch, and A. Kepecs, "Confidence and certainty: Distinct probabilistic quantities for different goals", *Nature neuroscience*, vol. 19, no. 3, pp. 366–374, 2016.

[9] J. I. Sanders, B. Hangya, and A. Kepecs, "Signatures of a statistical computation in the human sense of confidence", *Neuron*, vol. 90, no. 3, pp. 499–506, 2016.

[10] A. Zylberberg, P. Barttfeld, and M. Sigman, "The construction of confidence in a perceptual decision", *Frontiers in integrative neuroscience*, vol. 6, p. 79, 2012.

[11] A. Koizumi, B. Maniscalco, and H. Lau, "Does perceptual confidence facilitate cognitive control?", *Attention, Perception, & Psychophysics*, vol. 77, no. 4, pp. 1295–1306, 2015.

[12] B. Maniscalco, M. A. Peters, and H. Lau, "Heuristic use of perceptual evidence leads to dissociation between performance and metacognitive sensitivity", *Attention, Perception, & Psychophysics*, vol. 78, no. 3, pp. 923–937, 2016.

[13] A. Zylberberg, C. R. Fetsch, and M. N. Shadlen, "The influence of evidence volatility on choice, reaction time and confidence in a perceptual decision", *Elife*, vol. 5, e17688, 2016.

[14] M. A. Peters, T. Thesen, Y. D. Ko, B. Maniscalco, C. Carlson, M. Davidson, W. Doyle, R. Kuzniecky, O. Devinsky, E. Halgren, and H. Lau, "Perceptual confidence neglects decision-incongruent evidence in the brain", *Nature human behaviour*, vol. 1, no. 7, p. 0139, 2017.

[15] B. Odegaard, P. Grimaldi, S. H. Cho, M. A. Peters, H. Lau, and M. A. Basso, "Superior colliculus neuronal ensemble activity signals optimal rather than subjective confidence", *Proceedings of the National Academy of Sciences*, vol. 115, no. 7, E1588–E1597, 2018.

[16] J. Samaha and R. Denison, "The positive evidence bias in perceptual confidence is not post-decisional", *bioRxiv*, 2020.

[17] K. Miyoshi and H. Lau, "A decision-congruent heuristic gives superior metacognitive sensitivity under realistic variance assumptions.", *Psychological Review*, vol. 127, no. 5, pp. 655–671, 2020.

[18] D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, "Performance-optimized hierarchical models predict neural responses in higher visual cortex", *Proceedings of the national academy of sciences*, vol. 111, no. 23, pp. 8619–8624, 2014.

[19] A. J. Kell, D. L. Yamins, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott, "A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy", *Neuron*, vol. 98, no. 3, pp. 630–644, 2018.

[20] C. J. Cueva and X.-X. Wei, "Emergence of grid-like representations by training recurrent neural networks to perform spatial localization", in *International Conference on Learning Representations*, 2018.

[21] J. X. Wang, Z. Kurth-Nelson, D. Kumaran, D. Tirumala, H. Soyer, J. Z. Leibo, D. Hassabis, and M. Botvinick, "Prefrontal cortex as a meta-reinforcement learning system", *Nature neuroscience*, vol. 21, no. 6, pp. 860–868, 2018.

[22] B. Maniscalco and H. Lau, "A signal detection theoretic approach for estimating metacognitive sensitivity from confidence ratings", *Consciousness and cognition*, vol. 21, no. 1, pp. 422–430, 2012.

[23] L. Weiskrantz, "Blindsight: A case study and implications", 1986.

[24] N. Persaud, M. Davidson, B. Maniscalco, D. Mobbs, R. E. Passingham, A. Cowey, and H. Lau, "Awareness-related activity in prefrontal and parietal cortices in blindsight reflects more than superior visual performance", *Neuroimage*, vol. 58, no. 2, pp. 605–611, 2011.

[25] Y. Ko and H. Lau, "A detection theoretic explanation of blindsight suggests a link between conscious perception and metacognition", *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 367, no. 1594, pp. 1401–1411, 2012.

[26] C. G. Gross, "Contribution of striate cortex and the superior colliculus to visual function in area mt, the superior temporal polysensory area and inferior temporal cortex", *Neuropsychologia*, vol. 29, no. 6, pp. 497–515, 1991.

[27] J. A. Swets, "Form of empirical rocs in discrimination and diagnostic tasks: Implications for theory and measurement of performance.", *Psychological bulletin*, vol. 99, no. 2, p. 181, 1986.

[28] D. Rahnev, B. Maniscalco, T. Graves, E. Huang, F. P. De Lange, and H. Lau, "Attention induces conservative subjective biases in visual perception", *Nature neuroscience*, vol. 14, no. 12, pp. 1513–1515, 2011.

[29] D. A. Rahnev, B. Maniscalco, B. Luber, H. Lau, and S. H. Lisanby, "Direct injection of noise to the visual cortex decreases accuracy but increases decision confidence", *Journal of neurophysiology*, vol. 107, no. 6, pp. 1556–1563, 2012.

[30] A. Bhandari, C. Gagne, and D. Badre, "Just above chance: Is it harder to decode information from prefrontal cortex hemodynamic activity patterns?", *Journal of cognitive neuroscience*, vol. 30, no. 10, pp. 1473–1498, 2018.

[31] W. T. Adler and W. J. Ma, "Comparing bayesian and non-bayesian accounts of human confidence reports", *PLoS computational biology*, vol. 14, no. 11, e1006572, 2018.

[32] K. Khalvati, R. Kiani, and R. P. Rao, "Bayesian inference with incomplete knowledge explains perceptual confidence and its deviations from accuracy", *bioRxiv*, 2020.

[33] K. Xu, M. Zhang, J. Li, S. S. Du, K.-I. Kawarabayashi, and S. Jegelka, "How neural networks extrapolate: From feedforward to graph neural networks", in *International Conference on Learning Representations*, 2021.

[34] G. Okazawa, C. E. Hatch, A. Mancoo, C. K. Machens, and R. Kiani, "The geometry of the representation of decision variable and stimulus difficulty in the parietal cortex", *bioRxiv*, 2021.

[35] B. Maniscalco, B. Odegaard, P. Grimaldi, S. H. Cho, M. A. Basso, H. Lau, and M. A. K. Peters, "Tuned inhibition in perceptual decision-making circuits can explain seemingly suboptimal confidence behavior", *PLOS Computational Biology*, vol. 17, no. 3, pp. 1–28, 2021.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] M. N. Shadlen and W. T. Newsome, "Neural basis of a perceptual decision in the parietal cortex (area lip) of the rhesus monkey", *Journal of neurophysiology*, vol. 86, no. 4, pp. 1916–1936, 2001.

[38] J. I. Gold and M. N. Shadlen, "The neural basis of decision making", *Annu. Rev. Neurosci.*, vol. 30, pp. 535–574, 2007.

[39] H. F. Song, G. R. Yang, and X.-J. Wang, "Training excitatory-inhibitory recurrent neural networks for cognitive tasks: A simple and flexible framework", *PLoS computational biology*, vol. 12, no. 2, e1004792, 2016.

[40] M. Usher and J. L. McClelland, "The time course of perceptual choice: The leaky, competing accumulator model.", *Psychological review*, vol. 108, no. 3, p. 550, 2001.

4 Methods

4.1 Code

Simulations were carried out in Python using the following packages: PyTorch [41], NumPy [42], SciPy [43], scikit-learn [44], and Matplotlib [45]. All code is available on GitHub.

4.2 Datasets

Experiments were performed on three datasets. The first was the MNIST handwritten digits dataset, consisting of grayscale images of the digits 0-9 [46]. This dataset has a training set with $60,000$ images, and a test set with $10,000$ images. The second was the CIFAR-10 object classification dataset, consisting of color images of 10 common object categories (cats, dogs, cars, etc.) [47]. This dataset has a training set with $50,000$ images, and a test set with $10,000$ images. Third, we used an orientation discrimination task featuring oriented gabor patches. Images had a size of 32x32, and each image contained a single gabor patch, tilted either 5 degrees to the left or 5 degrees to the right. Gabor patches were generated using a centered Gaussian envelope with a standard deviation of 4 pixels, and a spatial frequency of 0.3 cycles per pixel.

For experiments using the MNIST and CIFAR-10 datasets, all networks were trained and evaluated on the official training and test sets respectively. Thus, for these datasets, all results presented in the paper involve generalization beyond the images used for training. Images from the MNIST dataset were resized from 28x28 to 32x32 using bilinear interpolation. For training on the CIFAR-10 dataset (with an original image size of 32x32), a random crop of size 32x32 was selected after zero-padding of size 4 on all sides, and images were flipped horizontally with a probability of 0.5.

During training, images were scaled by a contrast factor μ. This value was sampled online from a uniform distribution, and then multiplied by the values of each pixel (which had an original range of $[0, 1]$). Images were then normalized to the range $[-1, 1]$, and pixel-wise noise was added. Noise was sampled from a Gaussian distribution with variance σ. This value was also sampled online from a uniform distribution. After adding noise, images were thresholded to the range $[-1, 1]$ using a hard tanh function. For training on CIFAR-10, contrast was sampled from the range $[\mu = 0.1, \mu = 1]$ and noise was sampled from the range $[\sigma = 0.1, \sigma = 0.2]$. For training on the orientation discrimination task, contrast was sampled from the range $[\mu = 0.1, \mu = 1]$ and noise was sampled from the range $[\sigma = 0.5, \sigma = 1]$. For the standard MNIST training regime, contrast was sampled from the range $[\mu = 0.1, \mu = 1]$ and noise was sampled from the range $[\sigma = 1, \sigma = 2]$. Some experiments on MNIST used alternative training regimes, as described in Section 4.5.6.

4.3 Model architecture

The model architecture involved three major components. The first is a DNN encoder f_θ (with parameters θ) that takes an image $\boldsymbol{x}$, of class y, as input and outputs a low-dimensional embedding $\boldsymbol{z}$:

$$\boldsymbol{z} = f_\theta(\boldsymbol{x}) \tag{1}$$

This embedding is then passed to two output layers $g_{\phi_{class}}$ and $g_{\phi_{conf}}$. The layer $g_{\phi_{class}}$ (with parameters ϕ_{class}) outputs a predicted class $\hat{y}$ for the image:

$$\hat{y} = g_{\phi_{class}}(\boldsymbol{z}) \tag{2}$$

and the layer $g_{\phi_{conf}}$ (with parameters ϕ_{conf}) predicts $p(\hat{y} = y)$, the probability that the classification response is correct:

$$p(\hat{y} = y) = g_{\phi_{conf}}(\boldsymbol{z}) \tag{3}$$

The architectural details of these components depended on the specific datasets and experiments.

4.3.1 Encoder

For both handwritten digit classification (using the MNIST dataset) and orientation discrimination (trained with RL), f_θ consisted of 3 convolutional layers followed by 3 fully-connected (FC) layers. The convolutional layers had 32 channels each, a kernel size of 3, a stride of 2, batch normalization, and leaky ReLU nonlinearities with a negative slope of 0.01. The first 2 FC layers had 256 and 128 units, batch normalization, and leaky ReLU nonlinearities with a negative slope of 0.01. The low-dimensional embedding $\boldsymbol{z}$ was generated by the final FC layer, which had 100 units and no nonlinearity or normalization.

For experiments on CIFAR-10, a more challenging object classification benchmark, f_θ employed a more complex resnet architecture modeled closely on He et al. [48]. The basic building block of this architecture is the residual block, in which the input to a series of convolutional layers is added to their output, thus providing shortcuts in the computational graph that facilitate learning in very deep architectures. Our implementation of this component used the following formulation:

$$\tilde{\boldsymbol{x}}_b^{l=1} = ReLU(BN_b^{l=1}(conv_b^{l=1}(\boldsymbol{x}_{b-1}))) \tag{4}$$

$$\tilde{\boldsymbol{x}}_b^{l=2} = BN_b^{l=2}(conv_b^{l=2}(\tilde{\boldsymbol{x}}_b^{l=1})) \tag{5}$$

$$\boldsymbol{x}_b = ReLU(\tilde{\boldsymbol{x}}_b^{l=2} + \boldsymbol{x}_{b-1}) \tag{6}$$

where $\boldsymbol{x}_{b-1}$ is the output of the previous residual block $b-1$, $conv_b^{l=1}$ and $conv_b^{l=2}$ are the first and second convolutional layers in the current block b, and $BN_b^{l=1}$ and $BN_b^{l=2}$ are batch normalization layers. All convolutional layers employed a kernel size of 3 and no bias term. The default residual block employed layers with a stride of 1, and the same number of output channels as input channels. In some cases (as detailed in the next paragraph), a residual block had a different number of output than input channels, or employed a stride of 2 so as to output a smaller feature map. In these cases, the first convolutional layer $conv_b^{l=1}$ implemented either the stride or change in number of channels, and $\boldsymbol{x}_{b-1}$ was passed through an additional convolutional layer (incorporating the stride or change in number of channels) and batch normalization layer before being added to $\tilde{\boldsymbol{x}}_b^{l=2}$, to ensure that they had the same shape:

$$\tilde{\boldsymbol{x}}_b^{l=3} = BN_b^{l=3}(conv_b^{l=3}(\boldsymbol{x}_{b-1})) \tag{7}$$

$$\boldsymbol{x}_b = ReLU(\tilde{\boldsymbol{x}}_b^{l=2} + \tilde{\boldsymbol{x}}_b^{l=3}) \tag{8}$$

Residual blocks were further arranged into stacks. Each stack contained 9 residual blocks. Some stacks had a different number of output than input channels, implemented by the first block in the stack, and some stacks had an output stride of 2, implemented by the last block in the stack. The input image was first passed through an initial convolutional layer, with 16 channels, a stride of 1, batch normalization, and ReLU nonlinearities. This was followed by 3 residual stacks. The first stack had 16 output channels and an output stride of 2. The second stack had 32 output channels and an output stride of 2. The third stack had 64 output channels and an output stride of 1. Altogether, f_θ had 55 layers (1 initial layer + 3 stacks $\times$ 9 blocks per stack $\times$ 2 layers per block). Finally, $\boldsymbol{z}$ was generated by average pooling over the output of the third stack, yielding a 64-dimensional vector.

4.3.2 Output layers

For experiments on CIFAR-10 and the standard version of MNIST (both 10-way classification tasks), $g_{\phi_{class}}$ parameterized a categorical distribution indicating the predicted probability that $\boldsymbol{x}$ belonged to each of the 10 possible image classes, using a linear layer followed by a softmax nonlinearity. For the two-choice version of MNIST (classification of two randomly selected digit classes s1 and s2), $g_{\phi_{class}}$ parameterized a binomial distribution indicating the predicted probability that $\boldsymbol{x}$ belonged to class s2, using a linear layer followed by a sigmoid nonlinearity. For all experiments on CIFAR-10 and MNIST, $g_{\phi_{conf}}$ parameterized a binomial distribution indicating the predicted probability that the classification response was correct.

For the orientation discrimination task, we used an actor-critic architecture trained with RL. The actor and critic were separate output layers that both took $\boldsymbol{z}$ as input. The actor parameterized a categorical distribution over 3 possible actions (LEFT, RIGHT, and OPT-OUT) using a linear layer followed by a softmax nonlinearity. The critic predicted the reward that would be received on the current trial, using a linear layer.

4.4 Training

4.4.1 MNIST

For the standard 10-choice version of MNIST, $g_{\phi_{class}}$ was trained with a cross-entropy loss over the 10 possible image classes. For the two-choice version of MNIST, $g_{\phi_{class}}$ was trained with a binary cross-entropy loss. The target was 0 if $\boldsymbol{x}$ belonged to class s1 and 1 if $\boldsymbol{x}$ belonged to class s2. The confidence output layer $g_{\phi_{conf}}$ was trained with a binary cross-entropy loss. The target was 1 if the classification output was correct and 0 if the classification output was incorrect. The classification and confidence losses were summed, and the entire architecture was trained through backpropagation. Training was performed for 5 epochs using the Adam optimizer [49], with a learning rate of $5e-4$ and a batch size of 32. Note that for the two-choice version, these epochs were about 1/5 as long as they were for the standard version (since they only involved 2 out of the 10 possible image classes). All weights and biases were initialized using PyTorch defaults.

4.4.2 CIFAR-10

For the CIFAR-10 dataset, $g_{\phi_{class}}$ and $g_{\phi_{conf}}$ were trained with the same loss functions used for the standard 10-choice version of MNIST (cross-entropy loss for classification, binary cross-entropy loss for confidence), which were summed and used to train the entire architecture through backpropagation. Networks were trained for 164 epochs using stochastic gradient descent with weight decay of $1e-4$, momentum of 0.9, and a batch size of 128. An initial learning rate of 0.1 was used, which was then set to 0.01 at training epoch 82, and 0.001 at training epoch 123. All weights in f_θ were initialized using a Kaiming normal distribution [50], and all weights in the output layers $g_{\phi_{class}}$ and $g_{\phi_{conf}}$ were initialized using an Xavier normal distribution [51].

4.4.3 Reinforcement learning

For the orientation discrimination task, networks were trained using an actor-critic method [52]. During training, an action a_t was selected on each trial by sampling from the probability distribution generated by the actor. If LEFT or RIGHT was selected, the reward r_t received on that trial was 1 if the decision was correct and 0 if the decision was incorrect. If OPT-OUT was selected, the network received a smaller but guaranteed reward $r_{opt-out}$. This value was initialized to 0.5 at the beginning of training, and was then updated after each training batch i according to the following formula:

$$r_{opt-out_i} = min(accuracy_{i-1}, 0.75) \tag{9}$$

where $accuracy_{i-1}$ is the average accuracy for non-opt-out trials on the previous training batch. This setup prevented networks from defaulting to a strategy of always opting out early in training when accuracy was low.

Networks were trained to maximize reward in this task using the sum of two loss functions. The critic was trained using a smooth L1 loss to generate v_t, a prediction of the reward r_t received on that trial. This was then used to compute a reward prediction error:

$$\delta_t = r_t - v_t \tag{10}$$

which was used to compute a loss function for training the actor:

$$L_{actor} = -log(p_{a_t})\delta_t \tag{11}$$

where $-log(p_{a_t})$ is the negative log likelihood of the action sampled on that trial. The actor and critic losses were summed, and the entire architecture was trained through backpropagation. Networks were trained for 5000 iterations, using the Adam optimizer, with a learning rate of 0.001 and a batch size of 32. All weights and biases were initialized using PyTorch defaults.

4.5 Experiments and analyses

Unless otherwise noted, experiments were performed over 100 trained networks with different random initializations.

4.5.1 Meta-d'

To assess the metacognitive performance of our model, we used meta-d', a recently developed detection theoretic measure [22]. Just as d' measures the extent to which decisions discriminate between two stimulus classes, meta-d' measures the extent to which confidence ratings discriminate between correct and incorrect trials. Importantly, similar to d', meta-d' is not susceptible to response bias, i.e. the overall rate of high vs. low confidence ratings (except in extreme cases in which the decision-maker responds with either high confidence or low confidence on all trials). We used the python implementation available here for our analyses.

4.5.2 PE bias

To test for the presence of the PE bias, we evaluated networks at two different noise levels. For MNIST, we evaluated networks at $\sigma = 1$ and $\sigma = 2$; for CIFAR-10, we evaluated networks at $\sigma = 0.1$ and $\sigma = 0.2$; and for the orientation discrimination task, we evaluated networks at $\sigma = 0.5$ and $\sigma = 1$. For each noise level, we evaluated networks across a range of contrast levels. For MNIST and the orientation discrimination task, we evaluated networks at 500 values ranging from $\mu = 0$ to $\mu = 1$. For CIFAR-10, we evaluated networks at 500 values ranging from $\mu = 0$ to $\mu = 0.2$. For MNIST and CIFAR-10, networks were evaluated on the entire test set for each pair of σ and μ values. For the orientation discrimination task, networks were evaluated on $10,000$ trials for each pair of values.

We computed average decision accuracy across the networks trained on each task, for each pair of noise and contrast levels. For the orientation discrimination task, decision accuracy was computed by ignoring the OPT-OUT response, and selecting the argmax of the distribution over LEFT and RIGHT actions (rather than sampling from this distribution probabilistically as in training). For each noise level, we identified the contrast level with average decision accuracy closest to a target performance level. The target performance level was set to the threshold halfway between chance performance and 100% accuracy, since this is generally the most sensitive range for observing psychophysical effects. For MNIST and CIFAR-10, in which chance performance is 10% accuracy, the target performance was 55% accuracy. For the orientation discrimination task, in which chance performance is 50% accuracy, the target performance was 75% accuracy.

This procedure identified higher contrast values for higher noise values, and vice versa, resulting in a balanced signal-to-noise ratio across conditions. The procedure identified contrast values of $\mu = 0.27$ (for $\sigma = 1$) and $\mu = 0.54$ (for $\sigma = 2$) for MNIST, contrast values of $\mu = 0.08$ (for $\sigma = 0.1$) and $\mu = 0.16$ (for $\sigma = 0.2$) for CIFAR-10, and contrast values of $\mu = 0.3$ (for $\sigma = 0.5$) and $\mu = 0.48$ (for $\sigma = 1$) for the orientation discrimination task. We refer to the lower contrast / lower noise condition as the 'low PE' condition, and the higher contrast / higher noise condition as the 'high PE' condition. We computed the mean and standard error of decision accuracy for these two conditions, and also performed paired t-tests. This confirmed that decision accuracy was indeed balanced between the low and high PE conditions.

Finally, for MNIST and CIFAR-10, we compared confidence in the low and high PE conditions, by computing the mean and standard error over all networks trained on each task, and by performing paired t-tests. For the orientation discrimination task, we computed the opt-out rate in each condition, where opt-out trials were those on which the OPT-OUT response was the argmax of the distribution over all three actions. We computed the mean and standard error of the opt-out rate over all networks trained on this task, and performed paired t-tests.

4.5.3 Dissociation between type-1 and type-2 sensitivity

To test for the dissociation between type-1 and type-2 sensitivity identified by Maniscalco et al. [12], we used a two-choice version of the MNIST dataset. For each trained network, two randomly selected digit classes s1 and s2 were used. After training networks on this two-choice task, using images with the standard range of contrast and noise values (described in Section 4.2), networks were evaluated on five conditions. In each of these five conditions, images belonging to class s2 were presented at one of five contrast values $\mu_{i=1}$ through $\mu_{i=5}$. Images from class s1 were always presented at the intermediate contrast $\mu_{i=3}$. The noise level was set to $\sigma = 2$ in all conditions.

Contrast values were fit so as to reproduce the d' values observed in [12], using the procedure described in [35]. First, we fit the intermediate contrast $\mu_{i=3}$. To do so, we evaluated all trained networks on 200 contrast levels ranging from $\mu = 0.25$ to $\mu = 0.45$. For each contrast level, we evaluated networks on the entire test set (presenting both s1 and s2 at the same contrast) and computed the average d'. We then used linear interpolation to identify a contrast value corresponding to the target d' for this condition. This resulted in an intermediate contrast of $\mu_{i=3} = 0.36$. We then evaluated the networks again, presenting s1 at this intermediate contrast, and presenting s2 at a range of

1000 contrast levels from $\mu = 0$ to $\mu = 1$. For each contrast, we again evaluated networks on the entire test set, computed the average d', and used linear interpolation to identify contrast values corresponding to the target d' for the other conditions. This resulted in contrast values for those conditions of $\mu_{i=1} = 0.05$, $\mu_{i=2} = 0.16$, $\mu_{i=4} = 0.58$, and $\mu_{i=5} = 0.81$. We then evaluated networks on the entire test set for each of these five conditions, recording the trial-by-trial decision and confidence outputs generated by the networks.

Finally, we fit a type-2 noise parameter ξ, intended to model the additional accumulation of noise between the time at which decisions and confidence ratings are made. Type-2 noise was incorporated by mapping trial-by-trial confidence ratings into the range $[-\infty, \infty]$ using the logit function, adding Gaussian noise with variance ξ, and then mapping back to the range $[0, 1]$ using the sigmoid function. We computed meta-d' and response-specific meta-d' (meta-d' for s1 vs. s2 responses) for all five conditions, performing a grid search across 20 values from $\xi = 0.1$ to $\xi = 2$. We identified the type-2 noise level resulting in the closest fit to the observed meta-d' values, as measured by mean-squared error, resulting in a value of $\xi = 1.2$. We then used the fitted values for $\mu_{i=1..5}$ and ξ, and computed the mean and standard error (across all trained networks) of d', meta-d' and response-specific meta-d' in each of the five conditions. We also performed a version of this analysis without the additional noise term ξ (results presented in Section S2).

4.5.4 Decoding analyses

We applied the decoding analysis devised by Peters et al. [14] to our model in the following manner. First, we trained networks on the two-choice version of MNIST, using images with the standard range of contrast and noise values (described in Section 4.2). We then evaluated networks on the entire test set for each of 500 contrast levels between $\mu = 0.1$ and $\mu = 1$, with a noise level of $\sigma = 2$, and selected the contrast level that resulted in performance closest to a target accuracy of 75%. This procedure identified a contrast level of $\mu = 0.33$. We used images with that contrast level and a noise level of $\sigma = 2$ for training decoders.

For each trained network, we trained a decoder to predict the stimulus class of an image given the resulting state of the network. The input to the decoder was formed by concatenating the activation states for all layers in the network (using flattened versions of the convolutional layers), yielding inputs of size $11,236$. The decoder was itself a single neural network layer with a sigmoid nonlinearity, and no bias term. The decoder was trained using a cross entropy loss to predict the class y of an image $\boldsymbol{x}$ passed to the network. This training was performed for 5 epochs with the same training set used to train the network, using the ADAM optimizer with a learning rate of $5e-4$ and a batch size of 32.

We then simulated the ROC analyses from [14]. These analyses depend on having, for any given input image $\boldsymbol{x}$, an independent estimate of the neural evidence in favor of each stimulus class s1 and s2 (i.e. an estimate of the evidence in favor of s1 independent of the evidence in favor of s2, and vice versa). The following formulation was used to obtain these estimates:

$$e_{s1} = \frac{|\boldsymbol{w}_{\boldsymbol{w}<0}|\tilde{\boldsymbol{x}}_{\boldsymbol{w}<0}}{N_{\boldsymbol{w}<0}} \tag{12}$$

$$e_{s2} = \frac{\boldsymbol{w}_{\boldsymbol{w}>0}\tilde{\boldsymbol{x}}_{\boldsymbol{w}>0}}{N_{\boldsymbol{w}>0}} \tag{13}$$

where $\tilde{\boldsymbol{x}}$ is the concatenated state of all layers in the network following presentation of the image $\boldsymbol{x}$, $\boldsymbol{w}$ are the weights of the trained decoder, $\boldsymbol{w} < 0$ are indices for weights that are less than 0, $\boldsymbol{w} > 0$ are indices for weights that are greater than 0, $N_{\boldsymbol{w}<0}$ is the number of dimensions with decoder weights less than 0, and $N_{\boldsymbol{w}>0}$ is the number of dimensions with decoder weights greater than 0. Only images from the test set were used for these analyses.

These values were then used as inputs to four ROC analyses. For each analysis, an independent variable (IV) was generated by applying a decision rule to the neural evidence estimates, and both ROC curves and choice probability (area under the ROC curve) were computed to determine to what extent that IV could predict a binary dependent variable (DV). The first analysis determined to what extent the BE rule could predict the network's decisions. The IV was the balance-of-evidence, e_{s2} - e_{s1}, and the DV was the network's decision output (target = s2 decision). The second analysis determined to what extent the RCE rule could predict the network's decision outputs. For that analysis, two ROC curves were computed and averaged together: 1) using e_{s1} as an IV, and s1 decisions as targets, and 2) using e_{s2} as an IV, and s2 decisions as targets. The third analysis determined to what extent the BE rule could predict the network's confidence outputs. The IV was the difference between the evidence in favor

of the decision made and the evidence against the decision made, i.e. e_{s1} - e_{s2} if an s1 decision was made and e_{s2} - e_{s1} if an s2 decision was made, and the DV was the network's confidence outputs (target = confidence > 0.75). The fourth analysis determined to what extent the RCE rule could predict the network's confidence outputs. The IV was the response-congruent-evidence, i.e. e_{s1} if an s1 decision was made and e_{s2} if an s2 decision was made, and the DV was the the network's confidence outputs (target = confidence > 0.75). For each analysis, the average and standard error of the ROC curves and choice probability were computed across all trained networks. Additionally, paired t-tests were performed to determine whether there was a statistically significant difference between decision rules (BE vs. RCE) for either decisions or confidence, and whether there was a statistically significant interaction between decision rule and DV (decisions vs. confidence). Finally, we also carried out an alternative version of this analysis using only the activities from the penultimate layer $\boldsymbol{z}$ as input to the decoder (Supplementary Figure S3).

4.5.5 Blindsight

To test for the presence of blindsight-like effects, we used networks trained on the two-choice version of MNIST, using contrast values sampled from the range $[\mu = 0.1, \mu = 1]$ and noise values sampled from the range $[\sigma = 3, \sigma = 4]$. After training, we simulated a lesion to the first layer of the network by multiplying the activities in that layer by a scaling factor of 0.01. We then evaluated the network on the entire test set at five contrast levels from $\mu = 0.5$ to $\mu = 0.9$, with a noise level of $\sigma = 4$. For each contrast value, we computed the mean and standard error for both d' and meta-d' across all trained networks. We also computed density estimates, using a Gaussian kernel, for the network's confidence outputs for correct vs. incorrect trials. These measures were compared to control networks without a lesion.

4.5.6 Alternative training and test regimes

We compared the standard training regime (contrast sampled from the range $[\mu = 0.1, \mu = 1]$, noise sampled from the range $[\sigma = 1, \sigma = 2]$) to three alternative regimes. In the 'fixed μ' regime, contrast was set to a fixed value of $\mu = 0.5$, while noise was sampled from the standard range $[\sigma = 1, \sigma = 2]$. In the 'fixed σ' regime, contrast was sampled from the standard range $[\mu = 0.1, \mu = 1]$, while noise was set to a fixed value of $\sigma = 1.5$. In the 'fixed μ/σ' regime, contrast was sampled from the standard range $[\mu = 0.1, \mu = 1]$, and noise was set to a fixed ratio of the contrast level $\sigma = 3.75\mu$.

To test for the impact of these regimes on the presence of the PE bias, we trained networks on the 10-choice version of MNIST, sampling contrast and noise values according to one of the four regimes described above, and then followed the procedure described in Section 4.5.2 to test for the PE bias.

To evaluate the metacognitive performance of networks trained on these regimes, when tested on each of the regimes, we trained networks on the two-choice version of MNIST, sampling contrast and noise values according to one of the four regimes described above, and then evaluated networks on the entire test set using contrast and noise values sampled from each of the four regimes. For this experiment, we trained 200 networks on each regime, since statistically significant differences between some of the conditions could not be established with only 100 networks per regime. For each combination of training and test regime, we computed the average and standard error of meta-d' over all networks. For each test regime, we performed two-sample t-tests for meta-d' between networks trained on that regime vs. networks trained on each of the other regimes.

4.5.7 Analysis of learned decision rules

To better understand the decision rules learned by the network, we adapted a paradigm used by Koizumi et al. [11]. In that paradigm, participants were presented with superimposed oriented gratings, one tilted to the left and one tilted to the right, and asked to judge which grating had a higher contrast. To simulate this paradigm in our model, we first trained networks on the two-choice version of MNIST, using images with the standard range of contrast and noise values (described in Section 4.2). Then, we presented networks with images of two superimposed digits. Each image contained an instance of both stimulus classes the network was trained on. For instance, if a network was trained to discriminate the digits '7' vs. '2', we presented that network with images containing overlapping 7's and 2's. The contrast values for each digit class, μ_{s1} and μ_{s2}, were independently manipulated. We evaluated networks on the entire test set for 100 contrast values between $\mu = 0.1$ and $\mu = 1$, testing each combination of values for μ_{s1} and μ_{s2}, yielding $10,000$ combinations. After applying the separate contrast values to the image of each class, the two classes were superimposed according to the following formula:

$$\boldsymbol{x}_{combined} = max(\boldsymbol{x}_{s1}, \boldsymbol{x}_{s2}) \tag{14}$$

such that the value of each pixel in the combined image was the maximum value of that pixel for the image of each class. For evaluation, the noise level was set to a value of $\sigma = 1.5$. We treated the decision output of the network (trained to discriminate between classes s1 and s2) as the network's decision about which digit class had a higher contrast. We then computed, for each combination of contrast values μ_{s1} and μ_{s2}, the average decision accuracy and confidence over all trained networks.

We used regression models to formally compare the network's pattern of confidence behavior to the predictions of the BE and RCE models. We applied a logit function to transform the networks' confidence outputs from the range $[0, 1]$ to $[-\infty, \infty]$. We fit three linear regression models to predict this value. The first used the balance-of-evidence (BE $= \mu_{s2} - \mu_{s1}$) as a predictor. The second used the response-congruent-evidence (RCE $= \mu_{s1}$ if $\hat{y} = s1$, μ_{s2} if $\hat{y} = s2$) as a predictor. The third was a multiple regression incorporating both of these predictors. These regressions were fit for all trained networks, and we computed the average predictions for each regression across all combinations of μ_{s1} and μ_{s2}. To assess the fit of each regression, we computed the average and standard error of R^2 across all trained networks, and we compared the fit of different models by performing paired t-tests on their R^2 values.

4.5.8 Analysis of learned representations

To better understand the representations learned by the model, we applied principal component analysis (PCA) to the representations in the network's penultimate layer $\boldsymbol{z}$. We evaluated networks trained on the two-choice version of MNIST. Both training and evaluation used the standard range of contrast and noise values (described in Section 4.2). To assess the dimensionality of the learned representations, we computed the average and standard error of the variance explained by each principal component (PC) across all trained networks. This revealed that the variance was almost entirely explained ($> 97\%$) by the top two PCs alone. We computed density estimates, using a Gaussian kernel, for stimulus classes s1 and s2 along PC1, and density estimates for correct and incorrect trials along PC2. We also performed regression analyses to determine to what extent PCs1 and 2 predicted DVs of interest. We performed two logistic regressions: 1) P 1 as predictor and stimulus class s1 vs. s2 as DV, and 2) PC2 as predictor and correct vs. incorrect as DV. We performed three linear regressions: 1) |PC1| as predictor and PC2 as DV, 2) PC1 as predictor, and a logit transformation of the network's decision output $\hat{y}$ as DV, and 3) PC2 as predictor and a logit transformation of the network's confidence output as DV. For each regression, we computed the average and standard error for R^2 across all trained networks.

We also performed an analysis to determine whether variables other than the network's confidence output were subject to the PE bias. We applied the analysis described in Section 4.5.2 to three other DVs: 1) |PC1|, 2) PC2, and 3) a rectified version of the network's decision output, which was computed by taking the network's continuous decision output (a value in the range $[0, 1]$ representing the predicted probability that the input image belonged to class s2), and applying a rectification at 0.5, resulting in a value representing the predicted probability that the network assigned to its decision. We also performed an analysis to better understand the representations along PC1 for the low and high PE conditions. For each condition, we computed the difference in the mean along PC1 for s1 and s2 trials, and the average variance along PC1. We also computed density estimates for s1 and s2 trials, for the low and high PE conditions, along PC1. Finally, we applied the analysis described in Section 4.5.7, presenting images of two superimposed digits, and computing, for each combination of contrast levels, the average value for |PC1|, PC2, and the rectified decision output.

References

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library", in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.

[42] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy", *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[43] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python", *Nature Methods*, vol. 17, pp. 261–272, 2020.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[45] J. D. Hunter, "Matplotlib: A 2d graphics environment", *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[46] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[47] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images", 2009.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", in *International Conference on Learning Representations*, 2015.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[51] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[52] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems", *IEEE transactions on systems, man, and cybernetics*, no. 5, pp. 834–846, 1983.

Supplementary Figures

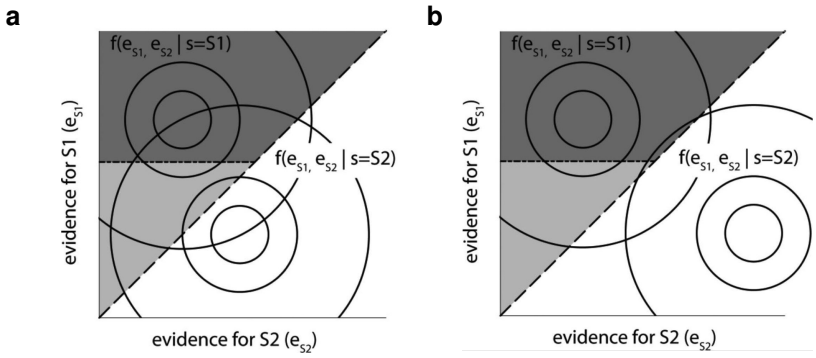

Figure S1: Graphical explanation of dissociation between type-1 and type-2 sensitivity. Maniscalco et al. [12] developed a paradigm to test for a dissociation between type-1 and type-2 sensitivity predicted by the RCE model. In that paradigm, participants detected whether a noisy grating was presented on the right or the left. One stimulus class, s1 (e.g. gratings on the left), was always presented at an intermediate contrast value, whereas the other stimulus class, s2 (e.g. gratings on the right), was presented at one of five contrast values. Under these conditions, the RCE and BE models make qualitatively distinct predictions. According to the BE model, both type 1 sensitivity (as measured by d') and type-2 sensitivity (as measured by meta-d') should increase as a function of the contrast value for s2, regardless of whether participants select s1 or s2. According to the RCE model, a dissociation is predicted for trials on which participants select s1 vs. s2. On trials with a response of s1, the RCE model predicts that type-2 sensitivity will go down as a function of contrast, whereas type-1 sensitivity will go up. On trials with a response of s2, the RCE model predicts that both type-1 and type-2 sensitivity will go up as a function of contrast. Panels **(a)** and **(b)** depict stimulus evidence distributions for two different conditions (figures reproduced from [12]). The horizontal axes represent evidence in favor of s2, e_{S2}, and the vertical axes represent evidence in favor of s1, e_{S1}. The distribution for s1 is represented by the concentric circles labelled $f(e_{S1}, e_{S2}|s = S1)$, and the distribution for s2 is represented by the circles labelled $f(e_{S1}, e_{S2}|s = S2)$. Panel **(a)** depicts the condition in which s2 is presented at a low contrast, and panel **(b)** depicts the condition in which s2 is presented at a high contrast, while s1 is presented at an intermediate contrast in both conditions. The area above the dashed diagonal line, in which there is more evidence for s1 than s2, corresponds to trials on which participants choose s1. According to the RCE model, confidence is a function of the response-congruent evidence only. Thus, in this region, confidence is exclusively a function of the evidence for s1. This is depicted by the dashed horizontal line separating high confidence responses (in the dark gray region) from low confidence responses (in the light gray region). As the contrast for s2 is increased, type-1 sensitivity increases, since the distributions for s1 and s2 become increasingly separated, resulting in a greater number of correct responses. However, because the movement in the distribution for s2 is only along the axis of evidence in favor of s2 (the horizontal axis), this increased sensitivity is not captured by confidence ratings that are only sensitive to evidence in favor of s1 (the vertical axis). Thus, s1 decisions in the light gray region are assigned low confidence, despite the fact that in panel **(b)** they are extremely unlikely to be incorrect (i.e. there is very little overlap between the distribution for s2 and the light gray region), resulting in lower metacognitive sensitivity relative to the scenario depicted in panel **(a)**.

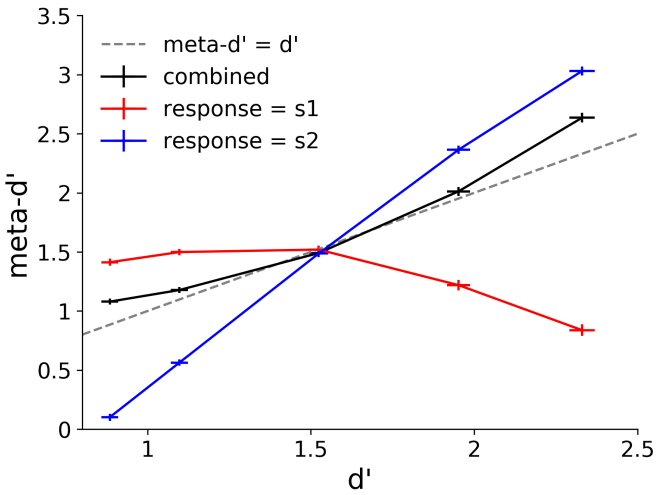

Figure S2: Dissociation between type-1 and type-2 sensitivity is present without additional type-2 noise. With no additional type-2 noise ($\xi = 0$), overall meta-d' is higher than in the original behavioral results, but the major qualitative effects are still present. The meta-d' vs. d' function has a negative slope for trials with a response of s1, and a positive slope for trials with a response of s2.

	Standard	Fixed μ/σ	Fixed μ	Fixed σ
Standard vs.		$p < 0.0001$	$p = 0.004$	$p < 0.0001$
Fixed μ/σ vs.	$p < 0.0001$		$p < 0.0001$	$p < 0.0001$
Fixed μ vs.	$p = 0.02$	$p = 0.003$		$p < 0.0001$
Fixed σ vs.	$p = 0.68$	$p < 0.0001$	$p = 0.006$	

Table S1: Statistical comparison of meta-d' for four test regimes. For each test regime, 200 networks trained on that regime (e.g. networks trained and tested on the standard regime) are compared to 200 networks trained in each of the other three regimes (e.g. networks trained on the fixed μ regime and tested on the standard regime). Each cell represents the results of a two-sample t-test.

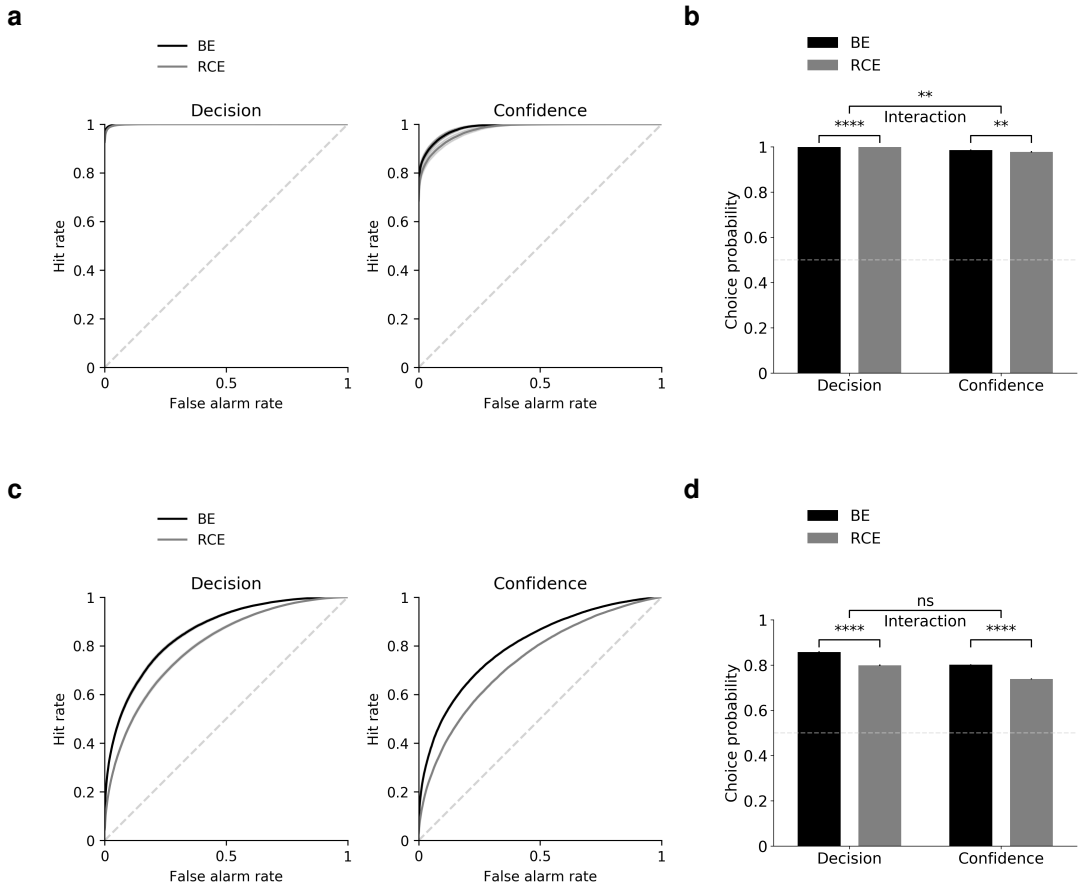

Figure S3: Decoding analysis in penultimate layer yields results consistent with presence of common decision variable. The decoding analysis described in Section 2.1.3, originally applied to a readout of the entire network, was applied here to the penultimate layer ($\boldsymbol{z}$) only. The original analysis indicated that decisions were better predicted by the balance of neural evidence, whereas confidence was equally well predicted by the balance of evidence and the response-congruent evidence alone. In contrast to those results, when the decoding analysis was applied to the penultimate layer only, the results were consistent with the presence of a common decision variable. Panels **(a)** and **(b)** show that both decisions and confidence were better predicted by the BE than the RCE rules, inconsistent with the results of the original decoding analysis. Furthermore, to the extent that an interaction was present, it was in the opposite direction of the interaction in the original results (the BE vs. RCE difference is greater for confidence than for decisions). However, we caution that these results are subject to a ceiling effect – decoding accuracy is very high for all conditions, due to the fact that the layer in question ($\boldsymbol{z}$) already represents the to-be-decoded variables in a linearly separable manner that can be read out by the subsequent output layers. To address this confound, we also performed a version of the decoding analysis in which additional noise (sampled from a Gaussian distribution with a variance of 2) was added to the activations of the penultimate layer before being passed to the decoder. This allowed us both to eliminate the ceiling effect, and to simulate the instrument noise present in real neural decoding analyses. Under these conditions (panels **(c)** and **(d)**), both decisions and confidence were better predicted by the BE than the RCE rules, and there was no interaction between decision rule and dependent variable (decisions vs. confidence), consistent with the presence of a common decision variable.