Adaptive Efficient Coding: A Variational Auto-encoder Approach

Guy Aridor
Columbia University
ga2449@columbia.edu

Francesco Grechi
Columbia University
fg2403@columbia.edu

Michael Woodford
Columbia University
mw2230@columbia.edu

Abstract

We study a model of neural coding with the structure of a variational auto-encoder. The model posits that the encoding of individual stimulus values is optimally adjusted for a finite training sample of stimuli retained in memory. We demonstrate that this model can rationalize existing experimental evidence on both perceptual discrimination thresholds and neural tuning curve widths in multiple sensory domains. Finally, since our model implies that encoding is optimized for a sample from the environment, it also provides predictions about the adaptation of neural coding as the environmental frequency distribution changes.

1 Introduction

An influential literature has proposed that variation in the degree of precision with which sensory magnitudes are encoded over the range of possible stimulus values can be explained by a principle of *efficient coding* [3, 4, 25], according to which a finite range of possible internal representations is used in a way that is well-adapted to the frequency distribution of the stimuli that the organism encounters in its environment. A variety of assumptions have been proposed as to the precise formulation of the relevant constraint on feasible encoding schemes, and the performance measure that an efficient coding scheme should maximize; under one popular proposal ("infomax" theories), the efficiency criterion should be maximal mutual information between the internal representation and the objective stimulus magnitude [11, 12, 16, 17, 28].

A relatively neglected topic in this literature has been the question of how an efficient internal representation scheme is supposed to be learned from experience of instances of individual stimuli. This question is relevant both for understanding how cortical maps self-organize during development, and for understanding the speed and reliability with which an efficient encoding scheme for a new frequency distribution should be expected to arise when the statistics of the environment change. In this paper we propose a statistical learning approach to neural coding that draws on recent work in unsupervised representation learning.

We posit that the way in which internal representations of sensory stimuli are formed and used in the nervous system has the structure of a *variational auto-encoder* (VAE) [14, 15]. Such a system includes both an encoding circuit, that produces a low-dimensional internal representation for any presented stimulus, and a decoding circuit that can produce a reconstructed value for the original stimulus on the basis of this coarse categorization. The decoder learns a generative model of the stimulus distribution in the environment, in which the category to which a stimulus is assigned plays the role of a latent explanatory variable; the encoder is assumed to be optimized to label stimuli in a way that will make the labeled data useful for training the decoder. This architecture is useful, not only because it provides an arguably realistic model of what encoding schemes are adapted to do well, but because it provides a model of how unsupervised learning of a set of coarse categories appropriate to a given environment can occur.

Preprint. Under review.

We show that this model of neural coding produces predictions regarding the widths of neural tuning curves and discrimination thresholds that are consistent with evidence from multiple sensory domains, so that it is competitive with other proposed models of efficient coding (e.g., [12]) in this respect. At the same time, our model provides an account of how an efficient coding scheme can be learned, and naturally allows the coding scheme to rapidly adapt to a new statistical environment, as we illustrate through a numerical example.

2 A Model of Neural Coding

2.1 General Setup

We suppose that each stimulus is described by a single real number x drawn independently from a continuous frequency distribution $\pi(x)$. Each stimulus is to be encoded as belonging to one of J latent categories, $\mathcal{J} = \{1, \ldots, J\}$; this bound on the number of possible categories is taken as a constraint. We consider neural coding systems with the structure of a VAE. In particular, we suppose that the nervous system learns an encoding rule that stochastically classifies a continuous stimulus as belonging to one of the discrete set of latent categories, with probabilities $p(j|x)$, and a decoding rule that stochastically decodes the latent category back to a continuous stimulus magnitude, with probabilities $\tilde{p}(x|j)$. Thus the perceived stimulus $\hat{x}$ is the stochastic output of the original input being encoded according to $p(j|x)$, and then decoded back to the stimulus space according to $\tilde{p}(\hat{x}|j)$. The collection of distributions $\{\tilde{p}(x|j)\}$, together with learned frequencies of occurrence $\{q(j)\}$ of the latent categories, form a generative model for the distribution of stimulus magnitudes in the environment; the distribution $\tilde{p}(x|j)$ can be thought of as a "posterior" distribution for the stimulus magnitude when a given stimulus is encoded using category j, in a model of approximate inference.

The encoding rule, or recognition model, must be chosen from a parametric family of possible rules, $p_\phi(j|x)$, where ϕ is a finite-dimensional vector of parameters. The encoding rule, combined with the environmental distribution $\pi(x)$, implies a joint distribution for true stimulus magnitudes and their labels given by

$$p_\phi(j, x) \;=\; \pi(x) \cdot p_\phi(j|x).$$

The decoding rule is likewise chosen from the family of parametric models, $p_\theta(x|j)$, where θ is a finite-dimensional vector of parameters. The implied generative model for the joint distribution of stimulus magnitudes and labels is then given by

$$\tilde{p}_\theta(j, x) \;=\; q_\theta(j) \cdot \tilde{p}_\theta(x|j),$$

where we include the frequencies $\{q(j)\}$ among the elements of θ. The problem for the recognition model is one of inferring the latent category j that has given rise to stimulus x, according to the generative model.

We can interpret this as a model of neural coding in a region of sensory cortex (say, the visual cortex), under the theory that the cortex maintains an internal generative model of how images are generated by underlying visual features, and that the role of early processing (for example, in V1) is then to invert this generative process and infer the extent to which a given image contains each of the possible features [18, 21, 23]. Following [9, 19], we suppose that each of our categories j represents a possible feature, and corresponds to a particular population of cortical neurons, with the rate of firing in each of the populations indicating an inferred posterior distribution over the possible features in the image. Under this interpretation, the conditional probabilities $p(j|x)$ in our model correspond to the relative firing rates of J populations of neurons. This allows us to derive quantitative predictions about the distribution of neural tuning curves, in addition to the model's predictions about the discriminability of different stimuli.

We suppose that the neural coding system is organized to learn a good representation of the environment. But as we do not assume that it is optimized for a particular downstream task, it remains a question what objective function the coding system ought to optimize; there has been considerable debate in the representation learning literature about which objectives lead to the most useful representations [5, 26]. A natural approach would be to follow [14, 15] and suppose that ϕ and θ are jointly optimized so as to minimize the Kullback-Leibler divergence of the joint distribution implied by the encoder relative to that implied by the decoder, $D_{KL}(p_\phi(j, x)||\tilde{p}_\theta(j, x))$. However, as noted by [7, 8], this would ensure a reasonable approximation to the environmental distribution $\pi(x)$, but would not necessarily lead to a meaningful latent representation. Instead, we follow [2], who propose

extending the objective function used in [14, 15] to explicitly incentivize the model to learn a more meaningful representation. Their "β-VAE" approach allows for more "disentangled" representations by providing an additional bonus for classification schemes in which the different categories are more informative about the underlying stimuli (the objective proposed in "infomax" theories).

Formally, we suppose that the parameters are optimized to solve the problem:

$$\min_{\phi,\theta} D + \beta R \tag{1}$$

D is a measure of the average distortion and R is a measure of the complexity resulting from the coding scheme. β trades off the relative importance assigned to minimizing distortion as opposed to complexity where a lower β implies a higher relative importance assigned to having informative categories. D and R are defined formally as follows:

$$D \equiv -\mathbb{E}_\pi\Big[\sum_j p_\phi(j \mid x) \log \tilde{p}_\theta(x \mid j)\Big]$$
$$R \equiv \mathbb{E}_\pi\Big[D_{KL}(p_\phi(j \mid x) \mid\mid q_\theta(j))\Big]$$

We suppose there is no restriction on the choice of $\{q_\theta(j)\}$. With unrestricted choice then it is clearly optimal, regardless of the value of β, to choose ϕ and θ such that $q_\theta(j) = p_\phi(j)$. Under this condition, (1) reduces to:

$$D + \beta R = D_{KL}(p_\phi(j,x) \mid\mid \tilde{p}_\theta(j,x)) + \underbrace{-\mathbb{E}_\pi\Big[\log \pi(x)\Big]}_{H} - (1-\beta)\underbrace{\mathbb{E}_\pi\Big[\sum_j p_\phi(j \mid x) \log \frac{p_\phi(j \mid x)}{p_\phi(j)}\Big]}_{I}$$

H represents the underlying entropy of the stimuli and is independent of ϕ or θ. I represents the Shannon mutual information between the category j and state x in the joint distribution produced by the recognition model. Thus, when $\beta < 1$, this objective assigns an additional bonus to classifications with higher mutual information between j and x, as desired. The smaller is β, the greater the emphasis placed on having more informative categories.

2.2 Training Data

We suppose that the parameters of the coding scheme are fit to a finite sample of observations drawn from the environment. Given a large set of previously observed stimuli, a sampling process selects a corpus of observations to be used in training the algorithm. We adopt a version of *reservoir sampling*, a standard algorithm from stream-processing [27], that provides a guarantee that, after any number of observations has been drawn, every previously observed stimulus has the same probability of being in the sample. Crucially, the standard version of reservoir sampling requires no knowledge of the total number of observations to be drawn and has a memoryless insertion and deletion policy, requiring only the current observation and the previous sample to be stored in memory at any given point.

We utilize a modification of traditional reservoir sampling, proposed in [1], that places greater weight on more recent observations over older observations. This temporal bias enables more rapid adaptation, as it allows for the possibility that the underlying environmental statistics are changing without requiring that the organism be explicitly aware of this shift. We utilize the version of reservoir sampling in [1] that results in an exponential bias in the sampling procedure but maintains the same desirable memoryless insertion and deletion policy. In particular, the algorithm has the property that the probability that the r-th observation being in the sample after the t-th observation is given by $f(r,t) = e^{-\lambda(t-r)}$. In this case, $\lambda = \frac{1}{m}$ where m is the overall memory size. Thus, we introduce a single additional parameter specifying the memory size m which will implicitly define the degree of temporal bias.[1]

[1] The details of the algorithm are displayed as Algorithm 1 in the supplementary material and the proof of its correctness is in [1]. Section 2 of the supplementary material further provides exercises demonstrating the nature of adaptation under this sampling scheme.

2.3 Parameterization and Learning Process

We illustrate our approach using a parameterization in which the generative model must be a finite mixture of Gaussians. In particular, we assume that for each $j \in \mathcal{J}$:

$$\tilde{p}_\theta(x \mid j) \sim \mathcal{N}(\mu_j, \sigma_j^2)$$

The parameters θ then consist of values $\{q_j, \mu_j, \sigma_j\}$ for each $j \in \mathcal{J}$, where the q_j represent the mixture coefficients. We further assume that the parametric family of possible recognition rules is optimally adapted to this family of generative models, in the sense that for any generative model θ, there exists a recognition rule $\phi = \theta$ that minimizes $D + \beta R$ (over all possible recognition rules).

Thus our parametric family of recognition rules is given by

$$p_\theta(j \mid x) = \frac{q_j \exp[-\frac{1}{\beta}[\log \sigma_j + \frac{1}{2}(\frac{x-\mu_j}{\sigma_j})^2]]}{\sum_{\tilde{j}} q_{\tilde{j}} \exp[-\frac{1}{\beta}[\log \sigma_{\tilde{j}} + \frac{1}{2}(\frac{x-\mu_{\tilde{j}}}{\sigma_{\tilde{j}}})^2]]}$$

where the possible values of ϕ correspond to possible values of θ. Note that this kind of recognition rule can be implemented by a competition between J populations of neurons, in which the probability of a neuron in population j firing first (resulting in classification of the stimulus as of type j) in the case of stimulus x is proportional to the height of tuning curve j at point x in the stimulus space, and the tuning curves are Gaussian in shape. With this interpretation, our model makes predictions not only about the discriminability of different stimuli, but also about the distribution of preferred stimuli and tuning curve widths in a neural population code.

The problem of fitting the parameters of our model to a training data set reduces to the familiar problem of fitting the parameters θ of a Gaussian Mixture Model, with the small modification that we minimize $D + \beta R$ rather than maximizing the likelihood. We use a training data set that is generated according to the procedure in subsection 2.2, and utilize an Expectation-Maximization (E/M) algorithm to fit the parameters of our model. For the implementation of the E/M algorithm we follow [6], but with modifications to the likelihood function that are required by the inclusion of β in $p_\phi(j \mid x)$. Note that when $\beta = 1$, our procedures become identical to those described in [6].[2]

3 Stimulus Encoding in a Stationary Environment

We first consider a stationary environment where the underlying stimulus distribution is fixed, and consider the results from our model when $\log X \sim \mathcal{N}(\mu, \sigma^2)$ with $\mu = 1$ and $\sigma = 1$, though the qualitative patterns we identify hold for other distributions.[3] Furthermore, we set $m = 10,000$ and show the results for this fixed memory size.

3.1 Properties of the Learned Model

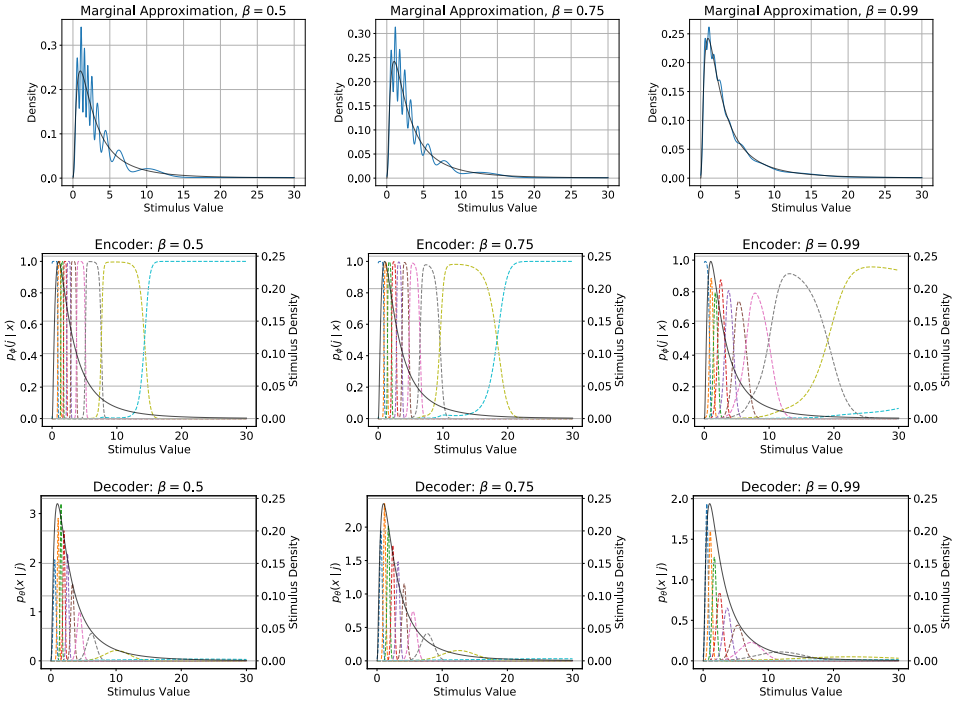
In this section we describe the qualitative properties of our neural coding model as we vary β and J. We first fix $J = 10$ and document the qualitative differences that result from varying β, and then proceed to analyze the case when J also varies. The qualitative patterns that emerge when varying β are robust to variation in J, and lead to significant qualitative differences in the resulting coding scheme.

We consider the grid of $\beta \in \{0.5, 0.75, 0.99\}$. The top row of Figure 1 shows the marginal distribution for x implied by the learned generative model (the VAE's approximation of π). Note that, as expected, when β gets closer to 1 the resulting marginal closely approximates the true π. As we lower β, the resulting marginal is a worse approximation to π. But this is expected, as a lower β leads the

[2] As noted by [13], poor initializations in the usage of the E/M may lead to convergence to "bad" local maxima. In order to ensure convergence, we first run the E/M algorithm initializing the μ_j values utilizing k-means. Then, we re-run the E/M algorithm with random initializations until the value of the objective function converges. Our numerical experiments showed that convergence occurs after 200 random initializations for the reported values of J, so that we utilize this for the results that follow.

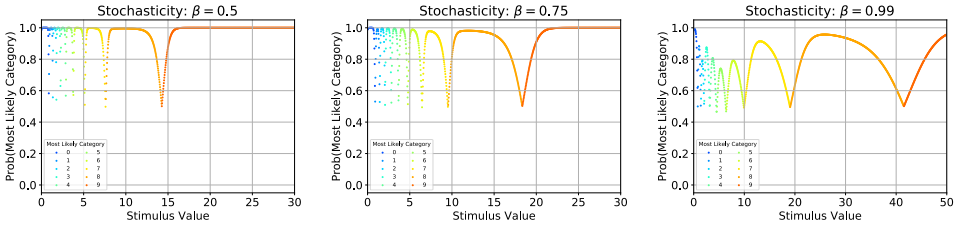
[3] Section 1 of the supplementary material reports similar exercises for different stimulus distributions. In addition, section 4 of the supplementary material considers alternative values of β and J than those considered here and shows the robustness of the qualitative patterns we document.

Figure 1: Marginal Approximation and Encoder / Decoder for $J = 10$, varying β



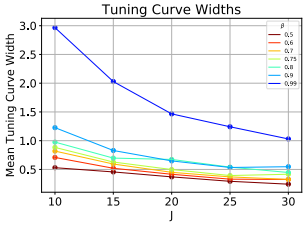
Notes: The top row plots the marginal distribution $p(x)$ implied by the generative model. The middle row plots, for each $j \in \mathcal{J}$, the recognition probability $p_\phi(j \mid x)$ across the support of π. The bottom row plots the $p_\theta(x \mid j)$ for each $j \in \mathcal{J}$.

Figure 2: Degree of Stochasticity for $J = 10$, varying β



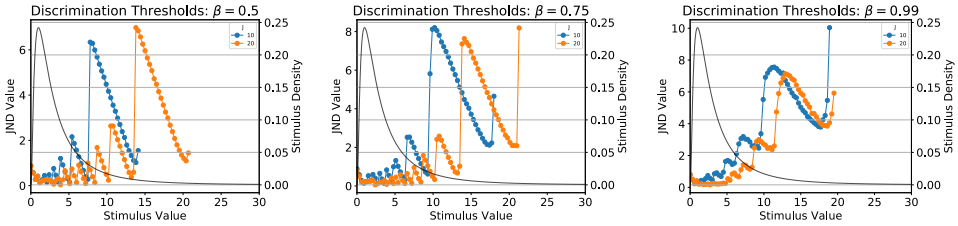
Notes: The figure displays a measure of the degree of stochasticity of the encoding. The y axis displays the probability that the encoder will encode a stimulus x using the category associated with the color. The curve plots $\max_j p_\phi(j \mid x)$ for each x, and the color at each point indicates $\arg\max_j p_\phi(j \mid x)$.

Figure 3: Tuning Curve Widths



Notes: The figure displays how the mean tuning curve width (computed by measuring the range of stimulus values for which the firing rate of a particular population of neurons is 50 percent or more of its maximum firing rate) decreases as J is increased.

Figure 4: Discrimination Thresholds for $J = 10, 20$, varying β



Notes: This figure displays the JND values computed for $c = 0.71$ and $J = 10, 20$. The resulting JND value is plotted as long as it exists. Note that for larger values of x, a JND value does not necessarily exist for a given c.

objective function to place more weight on having meaningful latent representations and less weight on ensuring a close approximation to the true π.

The middle and bottom rows of Figure 1 trace out $p_\phi(j \mid x)$ and $p_\theta(x \mid j)$, respectively, across all $j \in \mathcal{J}$ and $x \in supp(\pi)$. Several qualitative patterns are apparent. The first is that, contrary to models of neural coding where tuning curves are shifted versions of the same function [20, 24, 30, 31], in our model this is not generally the case. Instead, our model predicts that in regions of the stimulus space with high probability density, there is a higher density of neurons with narrower tuning curves whereas in portions of the stimuli space with lower probability mass there is a lower density of neurons with wider tuning curves.

There are also notable differences between the coding schemes obtained for different values of β. As β decreases, the encoding rule is increasingly deterministic, as is clear from Figure 1 and further apparent in Figure 2. Indeed, for $\beta = 0.5$ we find that the encoder is nearly deterministic except at stimulus values that are near category boundaries. However, for $\beta = 0.99$, there is considerable stochasticity in the optimal encoder.

Finally, we investigate the role that J plays in the resulting coding scheme. Sufficiently low values of J lead to poor approximations of π, even in the case that $\beta = 1$. Furthermore, numerical experiments confirm that increasing J weakly increases the value of the objective function. A natural question is how the coding scheme changes as we vary J. One possibility might be that as we increase J, the standard deviation of the various components does not change, but instead the components increasingly overlap. However, Figure 3 shows that as we increase J or decrease β, the mean σ and tuning curve width for the resulting models decrease. Thus, rather than having the same width components tiled across the stimulus space more densely, the components become narrower as J increases, so that their degree of overlap does not greatly increase.

3.2 Discriminability

In this section we apply our model of neural coding to generate predictions about stimulus discriminability. We characterize the qualitative differences in predictions as we vary β and J. In order to be qualitatively consistent with existing experimental evidence, our model should predict that the ability to discriminate between stimuli is inversely proportional to the frequency of occurrence associated with this stimuli in the environment [12].

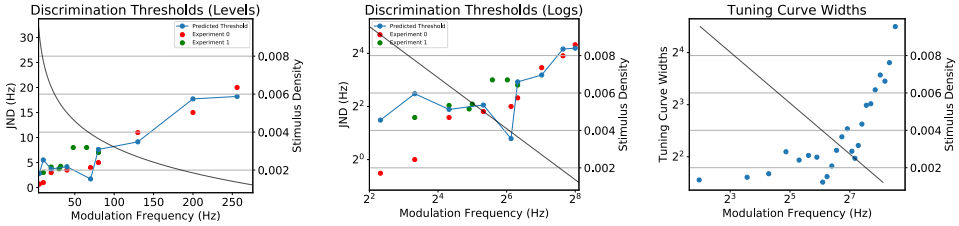
In order to study discriminability, we can define a *just noticeable differences* (JND) for each stimulus value x:

$$JND(c, x) = \underset{\Delta}{\operatorname{argmin}} \Pr(\widehat{(x + \Delta)} > \hat{x}) \geq c, \tag{JND}$$

where $\hat{x}$ is the random category to which a stimulus x is assigned, and $c > 0.5$. Figure 4 displays the resulting JND values for $c = 0.71$,[4] along with the density function π for comparison. It is apparent that regardless of the value of β, the qualitative predictions from the model are in line with experimental evidence, in that the predicted discrimination thresholds are lower for stimuli that occur more frequently in the environment.

[4] We report $c = 0.71$ in the main text since this is the value of c for which we report the calibration results in section 4. The supplementary material provides qualitatively similar plots for alternative values of c.

Figure 5: Discrimination Thresholds and Tuning Widths of Calibrated Model



Notes: The figures show the discrimination thresholds and neural tuning widths implied by the best fit model according to the calibration exercise ($J = 25$, $\beta = 0.62$). The figures on the left and in the middle display the experimentally observed discrimination thresholds, as well as the predicted discrimination thresholds of the calibrated model in levels and logs respectively. The figure on the right displays the predicted tuning widths for the calibrated model.

We observe in Figure 4 that as β increases, the JND values become smoother. Indeed, for lower β values we find a sawtooth pattern in the JND values, owing to the small overlap between successive components in this case, as already noted. Differentiation between two different stimuli requires that they be encoded as belonging to a different category; thus when categories are relatively discrete, discrimination thresholds are highest at the lower boundary of a category and gradually decrease as the upper boundary is approached. When the stimuli begin to be mapped into the next category, there is a jump in the threshold, leading to the sawtooth pattern.[5] Increasing β increases the JND curve's smoothness because it increases the amount by which the tuning curves overlap. This also reduces the model's ability to discriminate between nearby stimuli and thus results in larger JND values.

We also see that JND are decreasing in J for fixed β. Even for $\beta = 0.5$, we observe the same qualitative pattern for both $J = 10$ and $J = 20$, but the JND values are lower when $J = 20$. The mechanism behind the sawtooth pattern is the same, except that, as shown in Figure 3, the components are more concentrated as J increases. Increasing J narrows the width of the categories, resulting in smaller JND values.

4 Calibrating the Model to Experimental Evidence

In this section we compare the predictions of our model to an empirically observed stimulus distribution, the modulation frequency distribution reported by [12]. This is estimated from a compilation of animal vocalizations, background sounds, and recordings made while walking around a suburban university campus. Furthermore, [12] compile empirical evidence from existing studies of neural tuning widths and discrimination thresholds of organisms in this environment.[6]

We calibrate β and J to illustrate that the model can rationalize the experimentally observed discrimination thresholds. We choose β and J to solve the following problem:

$$\underset{\beta,J}{\operatorname{argmin}} \sum_{i \in I} \ell(disc^{OBS}(i), disc(i, \beta, J))$$

Here I denotes the set of experimental stimulus values, $disc^{OBS}(i)$ is the measured discrimination threshold for stimulus value i, and $disc(i, \beta, J)$ is the predicted discrimination threshold for stimulus i given parameter values β and J. The loss function ℓ penalizes discrepancies between the two values, and we consider three possible specifications.[7] Since it is computationally expensive to train the model for particular values of β and J, we search only over a discrete grid of possible values. We consider β in 0.01 increments from $[0.5, 0.95]$ and $J \in \{10, 15, 20, 25, 30, 35\}$. We

[5] Section 4, Figure 7 in the supplementary material provides evidence for this hypothesis.

[6] The neural tuning width data come from [22] and the data for perceptual discrimination thresholds come from [10, 29].

[7] The three loss functions that we consider are: $\ell(x, y) = |x - y|$, $\ell(x, y) = (x - y)^2$, or $\ell(x, y) = \frac{|x-y|}{x}$. These alternatives allow for comparisons based either on the absolute magnitude of the errors or the percentage deviation; the latter case allows a loss function that is independent of the magnitude of empirical discrimination thresholds.

pool the experimental results from the two separate studies in order to form the set I and the values $disc^{OBS}(i)$.

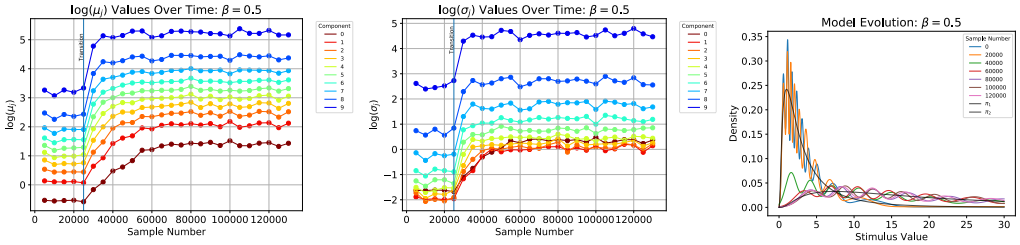
The resulting best fitting parameters for all three specifications of ℓ are $\beta = 0.62$ and $J = 25$. The observed and predicted discrimination thresholds as well as the underlying stimulus distribution are displayed in Figure 5. Figure 5 displays the results with the stimulus distribution presented in both logs and in levels, and shows that the calibrated model provides a reasonable fit to the measured discrimination thresholds in both experiments.

We further investigate whether the predicted tuning widths of the calibrated model are consistent with physiological data. While it is difficult to directly compare the predicted tuning widths to the physiological data, we verify that our model produces predictions that are qualitatively consistent with the observed data. In particular, we expect that the tuning widths should be approximately inversely proportional to the probability density at the "preferred stimulus" of the particular tuning curve. We define the width of the jth tuning curve as the length of the interval of stimulus values for which the tuning curve amplitude (firing rate) is at least half the amplitude at the peak (see the supplementary material for details). The resulting predicted tuning widths are displayed in Figure 5 and match these patterns. Overall, the model's predictions are in line with the experimental data on both discrimination thresholds and neural tuning widths.

5 Adaptation to a New Stimulus Frequency Distribution

In this section we demonstrate that the model also provides predictions about *adaptation* of the neural population code to a new environment. As an illustration, we consider a transition from $\pi_1 = \log X_1$, $X_1 \sim \mathcal{N}(\mu, \sigma^2)$, to $\pi_2 = \log X_2$, $X_2 \sim \mathcal{N}(\mu + \sigma, \sigma^2)$, with $\mu = 1$ and $\sigma = 1$ as before.

Figure 6: Adaptation of Parameters and Model



Notes: This figure depicts a parameter transition from $\log X \sim \mathcal{N}(\mu, \sigma^2)$ to $\log X \sim \mathcal{N}(\mu + 2 \cdot \sigma, \sigma^2)$. The figures on the left and center show the adaptation of the μ_j and $\log(\sigma_j)$ respectively. The figure on the right displays the evolution of the implied marginal distribution as more samples are drawn.

Figure 6 displays the adaptation of μ, σ, and the marginal distribution for x implied by the learned generative model, where the first $25,000$ samples are drawn from π_1, the remaining $105,000$ samples are drawn from π_2. The parameters of the model are eventually fully adapted to π_2 within this time period. The rate of convergence depends crucially on the memory size m, which is set to the same value as before ($m = 10,000$). In the supplementary material, we also present results for $m = 1,000$ and show that while the coding scheme adapts more rapidly with this lower memory size, it also induces additional jitter in the parameter values. Convergence occurs once the empirical distribution used to train the VAE fully transitions from a sample from π_1 to a sample from π_2, and this occurs faster with a lower m; but smaller m also leads to a less precise approximation of the distribution sampled from.[8]

A key advantage of our model of adaptation is that at no point does the perceptual system need to be instructed that the environment has changed. Rather, the sampling method that determines the training data set is constructed to rapidly adapt to a new environment owing to the temporally biased sampling. In future work we hope to explore how the quantitative predictions of such a model of adaptation in a neural population code match empirical evidence.

[8] In the supplementary material we also include an exercise showing how the memory size impacts the speed of convergence of the empirical distributions.

References

[1] AGGARWAL, C. C. On biased reservoir sampling in the presence of stream evolution. In *Proceedings of the 32nd international conference on Very large data bases* (2006), pp. 607–618.

[2] ALEMI, A., POOLE, B., FISCHER, I., DILLON, J., SAUROUS, R. A., AND MURPHY, K. Fixing a broken elbo. In *International Conference on Machine Learning* (2018), pp. 159–168.

[3] ATTNEAVE, F. Some informational aspects of visual perception. *Psychological review 61*, 3 (1954), 183.

[4] BARLOW, H. B., ET AL. Possible principles underlying the transformation of sensory messages. *Sensory communication 1* (1961), 217–234.

[5] BENGIO, Y., COURVILLE, A., AND VINCENT, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence 35*, 8 (2013), 1798–1828.

[6] BISHOP, C. M. *Pattern recognition and machine learning*. springer, 2006.

[7] BOWMAN, S., VILNIS, L., VINYALS, O., DAI, A., JOZEFOWICZ, R., AND BENGIO, S. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* (2016), pp. 10–21.

[8] CHEN, X., KINGMA, D. P., SALIMANS, T., DUAN, Y., DHARIWAL, P., SCHULMAN, J., SUTSKEVER, I., AND ABBEEL, P. Variational lossy autoencoder. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017).

[9] FISER, J., BERKES, P., ORBÁN, G., AND LENGYEL, M. Statistically optimal perception and learning: from behavior to neural representations. *Trends in cognitive sciences 14*, 3 (2010), 119–130.

[10] FORMBY, C. Differential sensitivity to tonal frequency and to the rate of amplitude modulation of broadband noise by normally hearing listeners. *The Journal of the Acoustical Society of America 78*, 1 (1985), 70–77.

[11] GANGULI, D., AND SIMONCELLI, E. P. Implicit encoding of prior probabilities in optimal neural populations. In *Advances in neural information processing systems* (2010), pp. 658–666.

[12] GANGULI, D., AND SIMONCELLI, E. P. Neural and perceptual signatures of efficient sensory coding. *arXiv preprint arXiv:1603.00058* (2016).

[13] JIN, C., ZHANG, Y., BALAKRISHNAN, S., WAINWRIGHT, M. J., AND JORDAN, M. I. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. In *Advances in neural information processing systems* (2016), pp. 4116–4124.

[14] KINGMA, D. P., AND WELLING, M. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings* (2014), Y. Bengio and Y. LeCun, Eds.

[15] KINGMA, D. P., WELLING, M., ET AL. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning 12*, 4 (2019), 307–392.

[16] LAUGHLIN, S. A simple coding procedure enhances a neuron's information capacity. *Zeitschrift für Naturforschung c 36*, 9-10 (1981), 910–912.

[17] LINSKER, R. Self-organization in a perceptual network. *Computer 21*, 3 (1988), 105–117.

[18] OLSHAUSEN, B. A., AND FIELD, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature 381*, 6583 (1996), 607–609.

[19] ORBÁN, G., BERKES, P., FISER, J., AND LENGYEL, M. Neural variability and sampling-based probabilistic representations in the visual cortex. *Neuron 92*, 2 (2016), 530–543.

[20] POUGET, A., DENEVE, S., DUCOM, J.-C., AND LATHAM, P. E. Narrow versus wide tuning curves: What's best for a population code? *Neural computation 11*, 1 (1999), 85–90.

[21] RAO, R. P., AND BALLARD, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience 2*, 1 (1999), 79–87.

[22] RODRÍGUEZ, F. A., CHEN, C., READ, H. L., AND ESCABÍ, M. A. Neural modulation tuning characteristics scale to efficiently encode natural sound statistics. *Journal of Neuroscience 30*, 47 (2010), 15969–15980.

[23] SCHWARTZ, O., AND SIMONCELLI, E. P. Natural signal statistics and sensory gain control. *Nature neuroscience 4*, 8 (2001), 819–825.

[24] SEUNG, H. S., AND SOMPOLINSKY, H. Simple models for reading neuronal population codes. *Proceedings of the National Academy of Sciences 90*, 22 (1993), 10749–10753.

[25] SIMONCELLI, E. P., AND OLSHAUSEN, B. A. Natural image statistics and neural representation. *Annual review of neuroscience 24*, 1 (2001), 1193–1216.

[26] TSCHANNEN, M., BACHEM, O., AND LUCIC, M. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069* (2018).

[27] VITTER, J. S. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS) 11*, 1 (1985), 37–57.

[28] WEI, X.-X., AND STOCKER, A. A. A bayesian observer model constrained by efficient coding can explain'anti-bayesian'percepts. *Nature neuroscience 18*, 10 (2015), 1509.

[29] WIER, C. C., JESTEADT, W., AND GREEN, D. M. Frequency discrimination as a function of frequency and sensation level. *The Journal of the Acoustical Society of America 61*, 1 (1977), 178–184.

[30] ZEMEL, R. S., DAYAN, P., AND POUGET, A. Probabilistic interpretation of population codes. *Neural computation 10*, 2 (1998), 403–430.

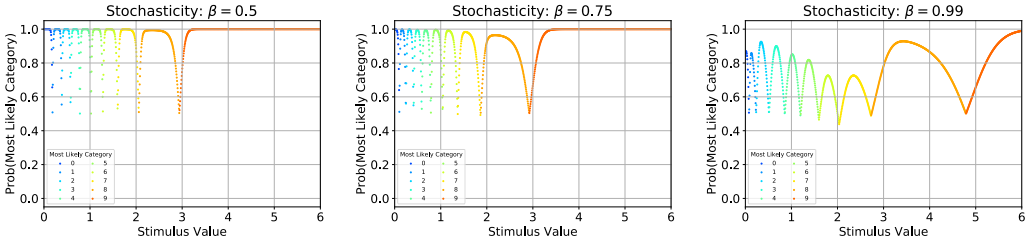
[31] ZHANG, K., AND SEJNOWSKI, T. J. Neuronal tuning: To sharpen or broaden? *Neural computation 11*, 1 (1999), 75–84.

Appendix for Adaptive Efficient Coding: A Variational Auto-encoder Approach

1 Figures for Alternative Stimuli Distributions

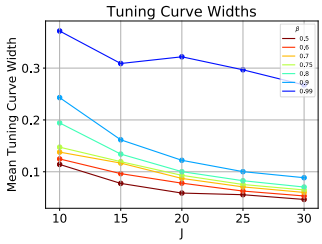
In this section we demonstrate that the qualitative insights from Section 3 are robust to the choice of the stimulus distribution. As an alternative, we here we consider the case $\pi \sim Exp(1.0)$ and reproduce the figures in Section 3 in the case of this alternative distribution.

Figure 1: Degree of Stochasticity for $J = 10$, varying β



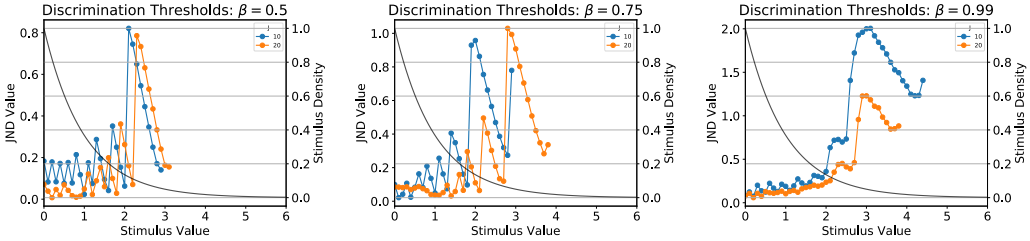
Notes: The figure displays a measure of the degree of stochasticity of the encoding. The y axis displays the probability that the encoder will encode a stimulus x using the category associated with the color. The curve plots $\max_j p_\phi(j \mid x)$ for each x, and the color at each point indicates $\arg\max_j p_\phi(j \mid x)$.

Figure 2: Tuning Curve Widths



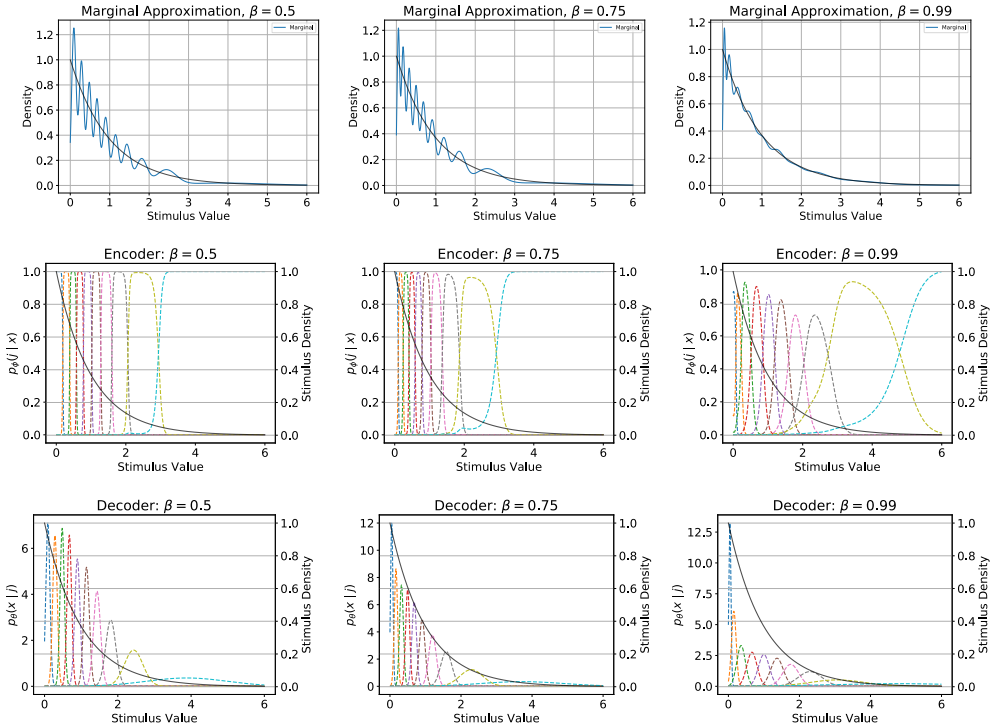
Notes: The figure shows how the mean tuning curve width (computed by measuring the range of stimulus values for which the firing rate of a particular population of neurons is 50 percent or more of its maximum firing rate) decreases as J is increased.

Figure 3: Discrimination Thresholds for $J = 10, 20$, varying β



Notes: This figure displays the JND values computed for $c = 0.71$ and $J = 10, 20$. The resulting JND value is plotted as long as it exists. Note that for larger values of x, a JND value does not necessarily exist for a given c.

Figure 4: Implied Marginal Distribution and Encoder / Decoder for $J = 10$, varying β



Notes: The top row plots the marginal distribution for x implied by the generative model. The middle row plots, for each $j \in \mathcal{J}$, $p_\phi(j \mid x)$ over the support of π. The bottom row plots instead $p_\theta(x \mid j)$ for each $j \in \mathcal{J}$.

2 Sample Adaptation

In this section we provide more details of the sampling procedure utilized to train the VAE in our model, and illustrate it via numerical examples. Algorithm 1 describes the details of the sampling algorithm from [1], a reservoir sampling algorithm with an exponential temporal bias. The algorithm guarantees that the probability that the r-th observed stimuli value is still present in the sample after the t-th observation is given by $f(r, t) = e^{-\lambda(t-r)}$, where $\lambda = \frac{1}{m}$ for a fixed memory size m. We focus on the case of adaptation following a transition from an original distribution π_1 to a new distribution π_2.

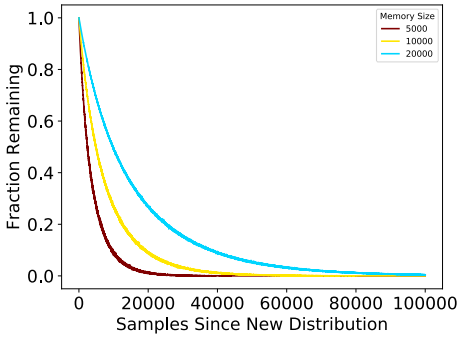
We first investigate the rate of decay of samples from the original π_1 distribution. Figure 5 displays the fraction of remaining observations from π_1 as more samples are drawn from π_2. The figure confirms that there is an exponential decay of samples from π_1, and that the rate of decay is faster for lower memory sizes. In the adaptation exercise in the main text, we use the value $m = 10,000$,

and find that after $40,000$ samples from π_2 the samples from π_1 have nearly vanished. We further demonstrate the resulting adaptation of the empirical distribution when $\pi_1 = \log X \sim \mathcal{N}(\mu, \sigma^2)$ and $\pi_2 = \log X \sim \mathcal{N}(\mu + \sigma, \sigma^2)$ as well as $\pi_2 = \log X \sim \mathcal{N}(\mu + 2 \cdot \sigma, \sigma^2)$ with $\mu = \sigma = 1.0$. Figure 6 displays the resulting empirical distribution during the transition between π_1 and π_2.

Algorithm 1 Temporally Biased Reservoir Sampling

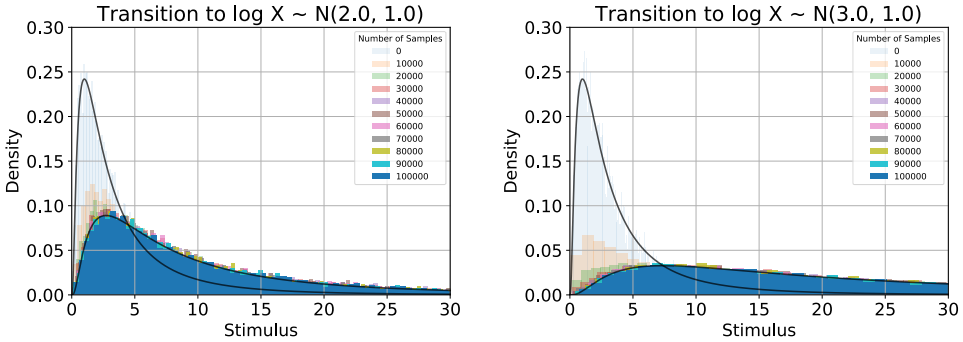
```
function BiasedReservoirSampling(draws,memory_size)
    reservoir = list()
    for item ∈ draws do
        F = length(reservoir) / memory_size
        if randomFloat(0, 1) < F then
            replaced_index = randomInteger(1, length(reservoir))
            reservoir[replaced_index] = item
        else
            reservoir.append(item)
        end if
    end for
    return reservoir
end function
```

Figure 5: Rate of Decay of Observations from π_1 in Memory



Notes: The figure displays the fraction of observations remaining in memory from π_1 after n samples are observed from π_2.

Figure 6: Empirical Distribution of Transition for $m = 10,000$



Notes: The figures show the empirical distribution in the training dataset following a change in the distribution π from which samples are drawn. The original distribution is $\log X \sim \mathcal{N}(1.0, 1.0)$, and in the figure on the left it transitions to $\log X \sim \mathcal{N}(2.0, 1.0)$; on the right it transitions to $\log X \sim \mathcal{N}(3.0, 1.0)$. The sampling procedure is as described in Algorithm 1.

3 Neural Tuning Curve Computation

In this section we provide additional details on the definition of neural tuning curves in the context of our model and provide a closed form calculation for the resulting neural tuning curve widths that are described in the main text.

3.1 Parametric families of functions used

We consider a finitely parameterized family of possible generative models,

$$\tilde{p}_\theta(j, x) \;=\; q_j \cdot \tilde{p}_\theta(x|j),$$

where the $\{q_j\}$ are also part of the vector θ. For any such generative model, the recognition model ϕ that will minimize $D + \beta R$ (allowing for a completely flexible recognition model) will be one such that

$$p_\theta(j|x) \;\sim\; q_j(\tilde{p}_\theta(x|j))^{\frac{1}{\beta}},$$

where we use the notation $\phi = \theta$ since the parameters of this model are same as those of the generative model for which the recognition model has been optimized. Note that if $\beta = 1$, the recognition model assigns conditional probabilities $p_\theta(j|x)$ that are equal to the conditional probability of a stimulus x having been produced as a draw from category j of the generative model, in the way that generative models are commonly used in Bayesian models of perception. However, when $\beta \neq 1$, this is no longer exactly the case.

In the case that the family of generative models considered is the family of finite mixtures of Gaussians, then the parameters are $\theta = \{q_j, \mu_j, \sigma_j\}$, and we have

$$\tilde{p}_\theta(x|j) \;\sim\; \frac{1}{\sigma_j} \exp[-\frac{1}{2}(\frac{x-\mu_j}{\sigma_j})^2],$$

$$p_\theta(j|x) \;\sim\; q_j \exp[-\frac{1}{\beta}[\log \sigma_j + \frac{1}{2}(\frac{x-\mu_j}{\sigma_j})^2]],$$

where in each case we have suppressed the common multiplicative factor required to make the conditional probabilities sum (or integrate) to 1.

3.2 A Neural Coding Interpretation

The recognition model $p_\phi(j|x)$ can be implemented by competition between pools of neurons in the following way. Suppose that there are J pools of neurons, with n_j neurons of each type j, and let x be a number on the real line indicating the physical magnitude of some stimulus feature. When a stimulus x is presented, each neuron of type j spikes at a Poisson rate proportional to $g_j(x)$, where the non-negative function $g_j(x)$ is the "tuning curve" for neurons of type j. We suppose that the stimulus is categorized as belonging to category j (i.e., is encoded by j) if the first spike is produced by a neuron of type j. Thus the recognition model implemented by the neural population is of the form

$$p_\phi(j|x) \;=\; \frac{n_j g_j(x)}{\sum_{\tilde{j}} n_{\tilde{j}} g_{\tilde{j}}(x)}.$$

The parametric family of recognition models that we assume in our version of a β-VAE are of this form, where

$$n_j g_j(x) \;\sim\; q_j(\tilde{p}_\theta(X|j))^{\frac{1}{\beta}},$$

and θ indicates the generative model for which the recognition model has been optimized. In the case that the generative model θ is a mixture of Gaussians parameterized by $\{q_j, \mu_j, \sigma_j\}$, we have

$$n_j g_j(x) \;\sim\; q_j \exp[-\frac{1}{\beta}[\log \sigma_j + \frac{1}{2}(\frac{x-\mu_j}{\sigma_j})^2]],$$

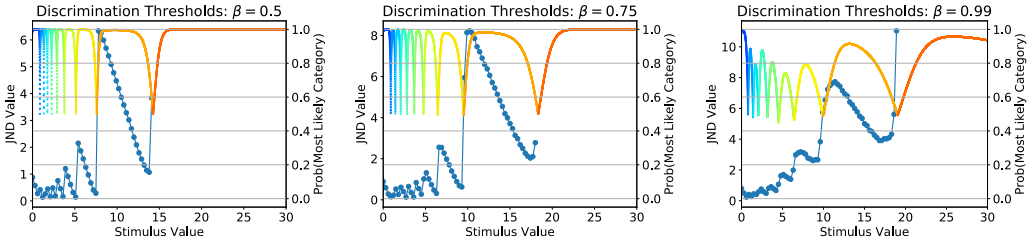
so that the tuning curve $g_j(x)$ must itself have a Gaussian shape (with standard deviation $\beta^{1/2}\sigma_j$) for each j. If we define the tuning curve width as the length of the interval of values $[\underline{x}_j, \bar{x}_j]$ over which $g_j(x) \geq (1/2)\max_{\tilde{x}} g_j(\tilde{x})$, then the tuning curve width for population j will equal $2\sqrt{2\beta \ln 2} \cdot \sigma_j$.

4 Additional Figures for Section 3

In this section we provide additional figures to complement the main analysis in section 3. Each figure is computed using the same stimulus distribution ($\log X \sim \mathcal{N}(1.0, 1.0)$) and memory size as in section 3. First, Figure 7 displays the relationship between the discrimination thresholds and the degree of stochasticity. Figure 7 validates the claim that the sudden increases in the discrimination thresholds align with transitions between adjacent categories, especially for lower values of β.

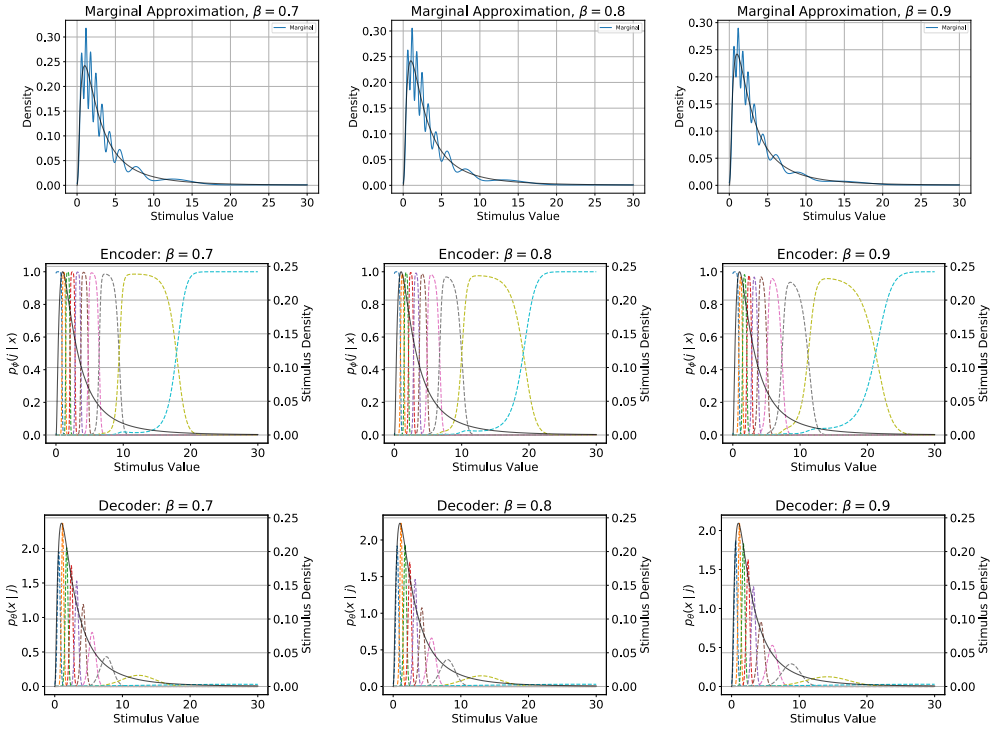
Figure 8 displays the resulting marginal approximation and corresponding encoder and decoder for the log normal distribution considered in the main text and with $J = 10$, but considers alternative β values to those considered in the main text. Figure 9 and Figure 10 display the marginal approximation and corresponding encoder and decoder for the same β values considered in the main text $\{0.5, 0.75, 0.99\}$ but for $J = 15$ and $J = 20$ respectively. Overall, these figures further validate the qualitative patterns described in the main text.

Figure 7: Discrimination Thresholds and Degree of Stochasticity, $J = 10$



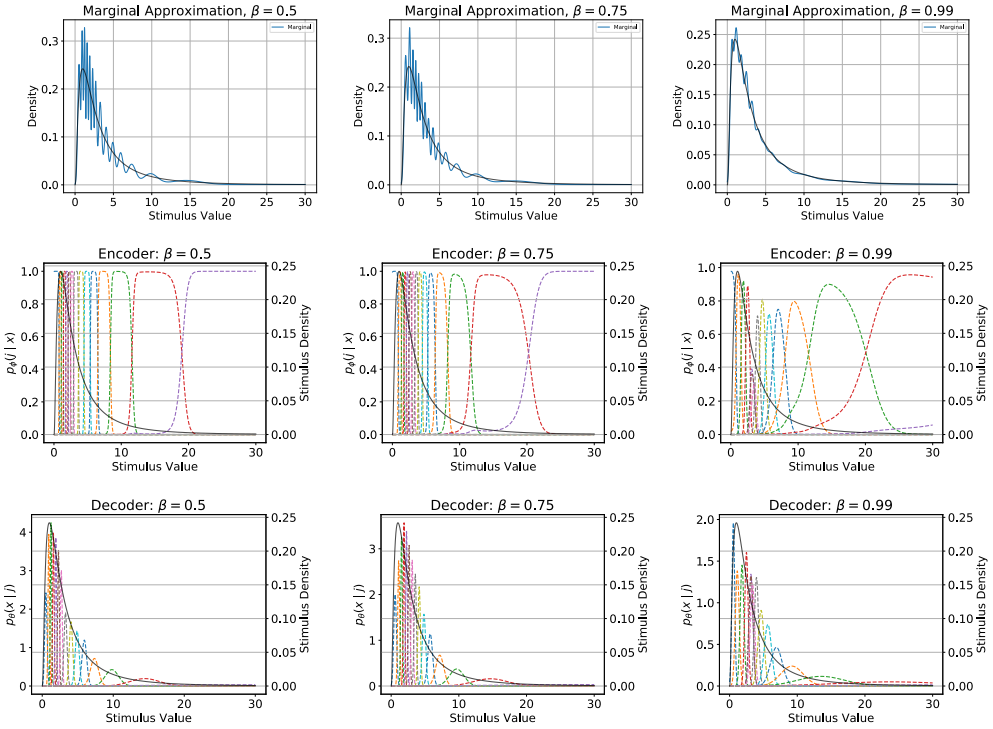
Notes: This figure displays the JND values paired with the degree of stochasticity measure for $\log X \sim N(1.0, 1.0)$ and for $J = 10$.

Figure 8: Marginal Approximation and Encoder / Decoder for $J = 10$, varying β



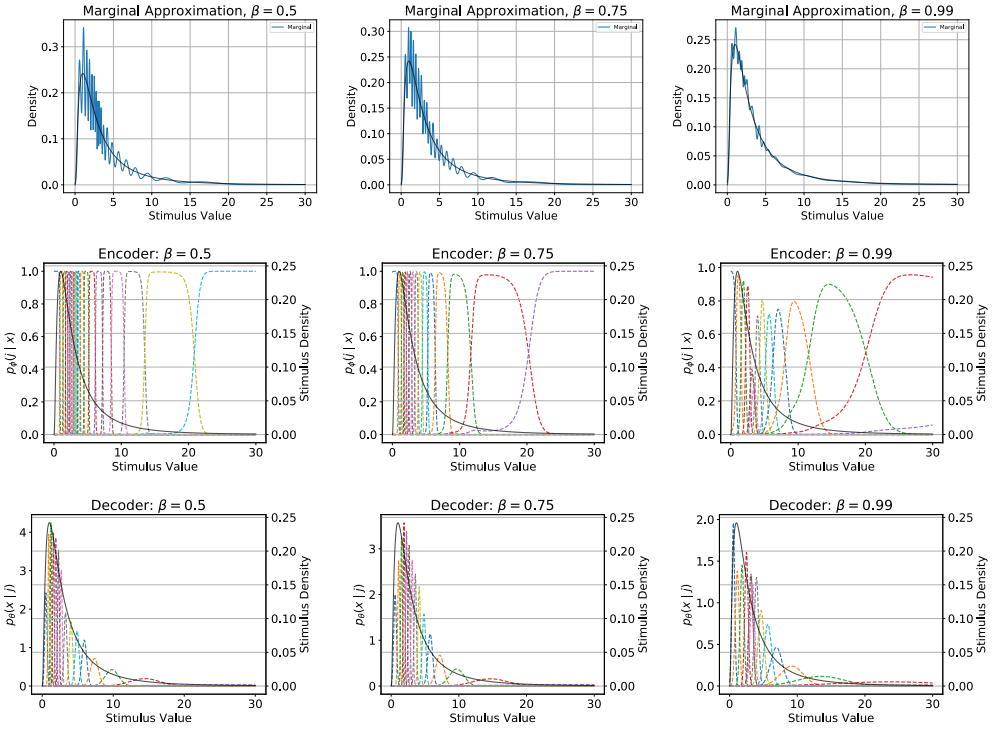
Notes: The top row plots the marginal approximation of π implied by the resulting model. The middle row plots, for each $j \in \mathcal{J}$, $p(j \mid x)$ across the support of π. The bottom row plots the $p(x \mid j)$ for each $j \in \mathcal{J}$.

Figure 9: Marginal Approximation and Encoder / Decoder for $J = 15$, varying β



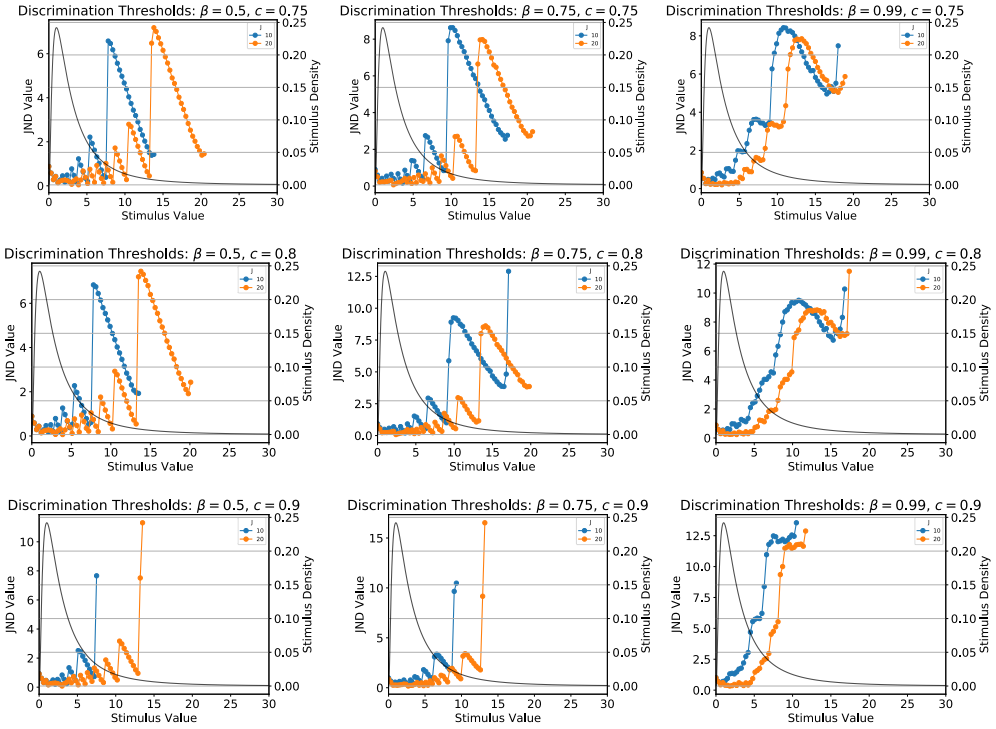
Notes: The top row plots the marginal approximation of π implied by the resulting model. The middle row plots, for each $j \in \mathcal{J}$, $p_\phi(j \mid x)$ across the support of π. The bottom row plots the $p_\theta(x \mid j)$ for each $j \in \mathcal{J}$.

Figure 10: Marginal Approximation and Encoder / Decoder for $J = 20$, varying β



Notes: The top row plots the marginal approximation of π implied by the resulting model. The middle row plots, for each $j \in \mathcal{J}$, $p_\phi(j \mid x)$ across the support of π. The bottom row plots the $p_\theta(x \mid j)$ for each $j \in \mathcal{J}$.

Figure 11: JND Plots for $c = 0.75, 0.8, 0.9$ and $J = 10, 20$, varying β

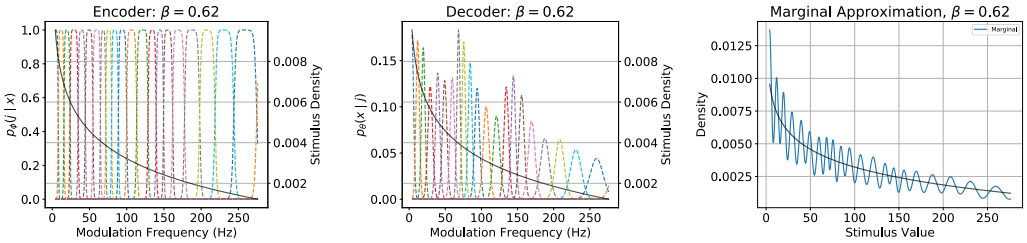


Notes: Each row plots the JND for $J = 10, 20$ and $\beta \in \{0.5, 0.75, 0.99\}$. The top row plots these values for $c = 0.75$, the middle row plots $c = 0.8$, and the last row plots $c = 0.9$.

5 Additional Figures for Section 4

In this section we provide additional figures for the calibration section. Figure 12 displays the resulting encoder, decoder, and implied marginal distribution for the calibrated parameters ($\beta = 0.62$, $J = 25$).

Figure 12: Encoding and Decoding Distribution Plots for Calibrated Discrimination Thresholds

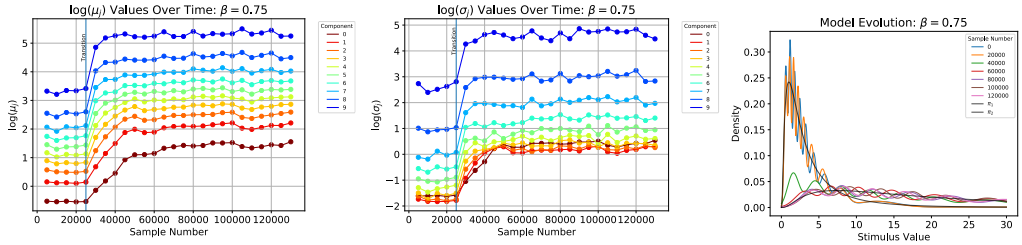


Notes: The figure displays the implied marginal distribution (right), encoder (left), and decoder (middle) for the calibrated coding scheme discussed in Section 4.

6 Additional Figures for Section 5

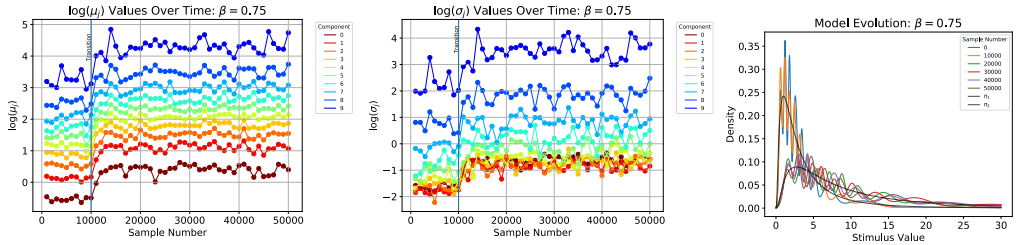
In this section we provide additional figures for the adaptation section. Figure 13 shows adaptation from $\pi_1 = \log X \sim \mathcal{N}(\mu, \sigma^2)$ to $\pi_2 = \log X \sim \mathcal{N}(\mu + 2 \cdot \sigma, \sigma^2)$ for $\beta = 0.75$. Figure 15 shows adaptation from $\pi_1 = \log X \sim \mathcal{N}(\mu, \sigma^2)$ to $\pi_2 = \log X \sim \mathcal{N}(\mu + \sigma, \sigma^2)$ for $\beta \in \{0.5, 0.75\}$ and further illustrates rapid adaptation to π_2. Finally, Figure 14 shows adaptation from $\pi_1 = \log X \sim \mathcal{N}(\mu, \sigma^2)$ to $\pi_2 = \log X \sim \mathcal{N}(\mu + \sigma, \sigma^2)$ for a significantly smaller memory size. Adaptation occurs more rapidly, but the resulting parameter estimates are considerably more noisy.

Figure 13: Adaptation of Parameters and Model, $\log X \sim \mathcal{N}(\mu + 2 \cdot \sigma, \sigma^2)$, $m = 10,000$



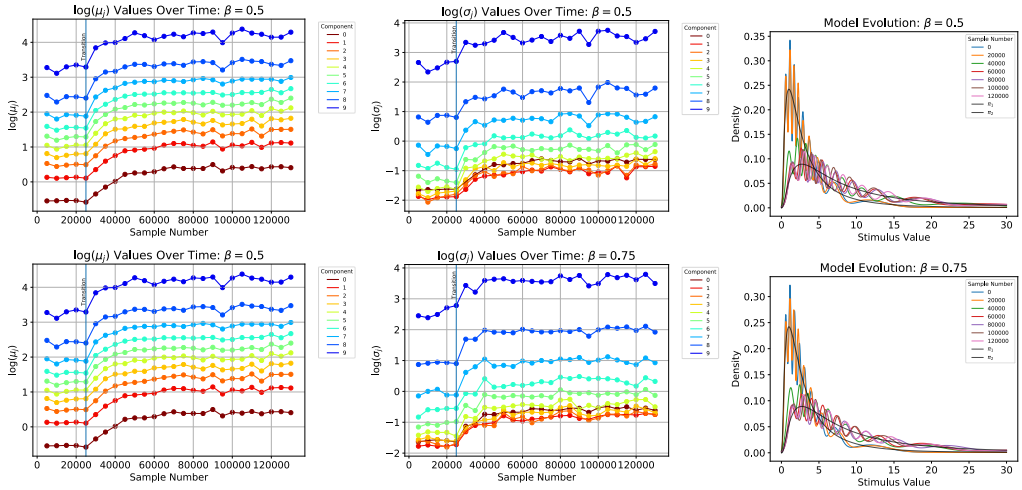
Notes: The figure shows adaptation of μ_j and σ_j as additional samples are drawn. The distribution shifts from $\pi_1 = \log X \sim \mathcal{N}(\mu, \sigma^2)$ to $\pi_2 = \log X \sim \mathcal{N}(\mu + 2 \cdot \sigma, \sigma^2)$ after 25,000 samples.

Figure 14: Adaptation of Parameters and Model, $\log X \sim \mathcal{N}(\mu + \sigma, \sigma^2)$, $m = 1,000$



Notes: The figure shows adaptation of μ_j and σ_j as additional samples are drawn. The distribution shifts from $\pi_1 = \log X \sim \mathcal{N}(\mu, \sigma^2)$ to $\pi_2 = \log X \sim \mathcal{N}(\mu + \sigma, \sigma^2)$ after 10,000 samples.

Figure 15: Adaptation of Parameters and Model, $\log X \sim \mathcal{N}(\mu + \sigma, \sigma^2)$, $m = 10,000$



Notes: The figure shows adaptation of μ_j and σ_j as additional samples are drawn. The distribution shifts from $\log X \sim \mathcal{N}(\mu, \sigma^2)$ to $\log X \sim \mathcal{N}(\mu + \sigma, \sigma^2)$ after 25,000 samples.