

A Double-Layer Multi-Resolution Classification Model for Decoding Spatiotemporal Patterns of Spikes With Small Sample Size

Xiwei She*

xiweishe@usc.edu

*Neuroscience Graduate Program, University of Southern California,
Los Angeles, CA, 90089, U.S.A.*

Theodore W. Berger

berger@usc.edu

Dong Song*

dsong@usc.edu

*Department of Biomedical Engineering and Neuroscience Graduate Program,
University of Southern California, Los Angeles, CA, 90089, U.S.A.*

We build a double-layer, multiple temporal-resolution classification model for decoding single-trial spatiotemporal patterns of spikes. The model takes spiking activities as input signals and binary behavioral or cognitive variables as output signals and represents the input-output mapping with a double-layer ensemble classifier. In the first layer, to solve the underdetermined problem caused by the small sample size and the very high dimensionality of input signals, B-spline functional expansion and $L1$ -regularized logistic classifiers are used to reduce dimensionality and yield sparse model estimations. A wide range of temporal resolutions of neural features is included by using a large number of classifiers with different numbers of B-spline knots. Each classifier serves as a base learner to classify spatiotemporal patterns into the probability of the output label with a single temporal resolution. A bootstrap aggregating strategy is used to reduce the estimation variances of these classifiers. In the second layer, another $L1$ -regularized logistic classifier takes outputs of first-layer classifiers as inputs to generate the final output predictions. This classifier serves as a meta-learner that fuses multiple temporal resolutions to classify spatiotemporal patterns of spikes into binary output labels. We test this decoding model with both synthetic and experimental data recorded from rats and human subjects performing memory-dependent behavioral tasks. Results show that this method can effectively avoid overfitting and yield accurate prediction of output

*Corresponding authors.

labels with small sample size. The double-layer, multi-resolution classifier consistently outperforms the best single-layer, single-resolution classifier by extracting and utilizing multi-resolution spatiotemporal features of spike patterns in the classification.

1 Introduction

Decoding neural signals is one of the fundamental problems in neuroscience and neural engineering (Brown, Barbieri, Ventura, Kass, & Frank, 2002; Marmarelis & Orme, 1993; Pouget, Dayan, & Zemel, 2003). It allows us to (1) understand the relationship between neural activities and external stimuli or behaviors (Donoghue, 2002; Esfahany, Siergiej, Zhao, & Park, 2018; Paninski, Pillow, & Simoncelli, 2004; Taylor, Helms Tillery, & Schwartz, 2003), (2) identify the functional connectivity between brain regions (Song et al., 2013; Stevenson et al., 2009), and (3) develop neural code-based prostheses for restoring sensory, motor, and cognitive functions (Carmenta et al., 2003; Collinger et al., 2013; Hampson et al., 2018; Hochberg et al., 2006; Humayun et al., 2003; Mauritz & Peckham, 1987; Nicolelis, 2003; Schwartz, Kettner, & Georgopoulos, 1988; Song & Berger, 2014). Although it is well known that the brain represents and processes information in the form of spatiotemporal patterns of spikes (Deadwyler & Hampson, 1995; Georgopoulos, Schwartz, & Kettner, 1986; Gerstner & Kistler, 2002; Hawkins & Blakeslee, 2004; Puchalla, Schneidman, Harris, & Berry, 2005; Schwartz et al., 1988), decoding single-trial spatiotemporal patterns of spikes remains a challenging task (Ghosh & Maunsell, 2021; Williams & Linderman, 2021).

Spike decoding can be formulated computationally as building an input-output model that takes spike trains as inputs and the signal to be decoded as output. Depending on the characteristics of the output signal, such decoding models can be categorized into regression models and classification models (see Figure 1). In regression models, output signals are continuous time-series variables such as 2D/3D kinematics (Donoghue, 2002; Nicolelis, 2003; Shanechi et al., 2013), brain states (Raposo, Kaufman, & Churchland, 2014; Shanechi et al., 2012), and spike firing probabilities (Qian et al., 2020; Song et al., 2007). In classification models, output signals are discrete categorical variables such as image labels (Quiroga, Reddy, Koch, & Fried, 2007; Song, She, Hampson, Deadwyler, & Berger, 2017; Zhang et al., 2020), discrete locations, and behavioral events (Cheng, Li, Fan, Shang, & Wan, 2021; Deadwyler, Bunn, & Hampson, 1996; Deadwyler & Hampson, 1995; Heelan et al., 2019; Premchand et al., 2020; Song et al., 2014; Sussillo et al., 2012; Warland, Reinagel, & Meister, 1997).

In the context of single-trial spike pattern analyses, it is often the case that classification models are more challenging than regression models to build since classification models often take spatiotemporal patterns with

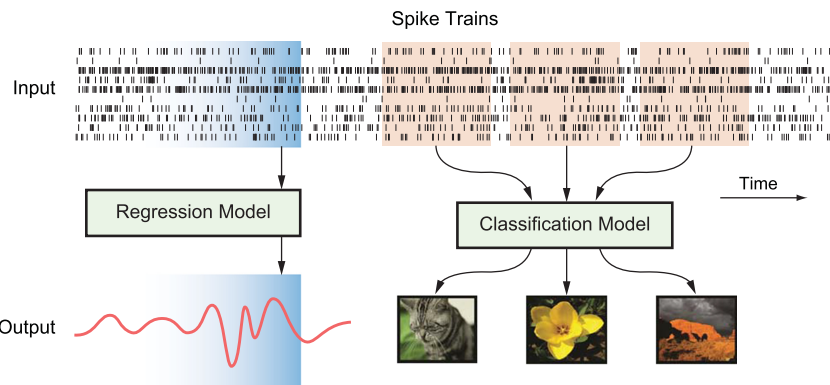


Figure 1: Decoding neural spikes with a regression model and a classification model. In a regression model, output is a continuous time-series variable. In a classification model, output is a discrete categorical variable.

longer durations of data samples and thus result in smaller sample sizes compared to regression models, which often take individual time points as data samples. Specifically, first, spatiotemporal patterns of spikes are extraordinarily high-dimensional signals to decode. Taking spike timings as features for decoding, the dimension of spatiotemporal patterns is equal to the product of the number of neurons and the number of possible spike timings within the decoding window, which can easily be on the order of thousands (Hampson et al., 2018; Song et al., 2017). Second, although the feature dimension can be reduced by lowering the temporal resolution of spike trains through binning, the optimal temporal resolution is not known a priori. Using a resolution that is too high cannot sufficiently reduce the feature dimension, while using a resolution that is too low will lose the temporal structure of spike patterns. Numerous studies have shown that precise spike timings are crucial in neural coding (Paugam-Moisy & Bohte, 2012; Thorpe, Delorme, & Van Rullen, 2001; Vreeken, 2002); different temporal resolutions are involved in different neurons, brain regions, and species (Arabzadeh, Panzeri, & Diamond, 2006; Lalor & Foxe, 2010; Oram, Xiao, Dritschel, & Payne, 2002). Third, output signals representing behaviors and cognitive states are also high dimensional. Taking the visual memory as an example, a single image can be labeled with numerous categories and features (Krizhevsky, Sutskever, & Hinton, 2017), leading to output signals also with numerous dimensions. Finally, classification models often suffer from having limited data length. Different from time-series regression models, which take continuous time steps with fine resolutions and thus result in a large number of samples (e.g., T), classification models are often estimated with discrete data epochs that each contains many data points and result in much a smaller sample size (e.g., T/M , where M is the epoch length).

For example, the numbers of behavioral events recorded from animals or human subjects ranging from 50 to 150 in data sets used in this study due to the time constraints of experimental or clinical procedures. Using such short data length to build classification models with high input and output dimensions poses a highly underdetermined estimation problem and becomes virtually impossible with conventional methods.

To solve these problems, we use modern machine learning techniques such as bootstrap aggregating (bagging), stacking, and regularized estimation to develop a double-layer, multi-resolution classification model for decoding spatiotemporal patterns of spikes. The model takes spatiotemporal patterns of spikes as input signals and behavioral and cognitive variables as output signals and represents the input-output mapping with a double-layer ensemble classifier. Binary labels are used as output signals to reduce the dimensionality of output space.

In the first layer, to solve the underdetermined problem caused by the small sample size and the high input and output dimensions, B-spline basis function (Smith & Schumaker, 1982) is used to reduce feature dimensionality; $L1$ -regularized (LASSO) logistic classifier is used to yield sparse model estimation. A wide range of temporal resolutions of neural signals is included by using a large number of classifiers with a different number of B-spline interior knots. Each classifier serves as a base learner to classify spatiotemporal patterns into the probability of a certain output label with a single temporal resolution. A bagging strategy is used to reduce estimation variances of these base learners by resampling the data and estimating an ensemble learner (Breiman, 1996). In the second layer, using a stacking strategy (Dietterich, 2000), another LASSO classifier takes outputs of all first-layer classifiers as inputs to yield the final output predictions. This second-layer classifier serves as a meta-learner that combines multiple temporal resolutions to classify spatiotemporal patterns of spikes into binary output labels. Nested cross-validation (Varma & Simon, 2006) is implemented in both layers to avoid overfitting and yield true out-of-sample estimations.

We test the proposed decoding method with both synthetic and experimental data recorded from rats and human subjects performing memory-dependent behavioral tasks. In experimental data, binary location labels and memory labels scored by normal human subjects are used as output signals in rat decoding and human decoding, respectively. Results show that this method can effectively avoid overfitting and yield accurate prediction of output labels. The double-layer, multi-resolution classifier consistently outperforms the best single-layer, single-resolution classifier by extracting and utilizing multi-resolution spatiotemporal features of spike patterns in the classification. Moreover, this decoding model derives a sparse classification functional matrix which provides characteristic spatiotemporal maps for decoding spatiotemporal patterns.

Table 1: Table of Notations.

Symbols	Meanings
η	Knot of B-spline basis functions
θ	Estimated probabilities
B	B-spline basis functions in units of probability
D	Data set pair $\{x, y\}$
F	Sparse classification functional matrix
G	Data set after bagging
H	Input-output decoding models
I	Number of instances
J	Number of B-spline basis functions
L	First-layer base learners
L'	Second-layer meta-learner
M	Length of the decoding window
N	Number of input signals
Q	Number of included B-spline resolutions
R	Number of Bagging replicas
S	Model estimation minimizer
b	B-spline basis function
d	Degree of B-spline basis function
l	Log-likelihood function
m	Number of interior knots of B-splines (B-spline resolutions)
w	Model coefficients
x	Input signals
y	Output signals
z	Input feature vectors
λ	Regularization parameter

2 Methods

2.1 Spike Decoding Model. Consider a spatiotemporal pattern of neural spikes X recorded around the i th trial of the behavioral event that happened at t_i , and that event is labeled as y . The goal is to build a model H that approximates the functional relationship between input x and output y :

$$H \left[X(t_i - \tau), -\frac{M}{2} \leq \tau \leq \frac{M}{2} \right] \rightarrow y(t_i) \quad (2.1)$$

where $X(t_i) = [x^{(1)}(t_i), x^{(2)}(t_i), \dots, x^{(N)}(t_i)]$, N is the number of input neurons, and M is the length of the decoding time window. With a sufficiently small sampling interval (e.g., 2 msec), x only contains binary values (1 when there is a spike, 0 when there is not) and preserves the temporal structures of spike trains. Notations used in this letter are summarized in Table 1.

Taking a probabilistic approach, equation 2.1 can be expressed as

$$\theta(i) = \text{Prob} \left(y(i) \middle| X(t_i - \tau), -\frac{M}{2} \leq \tau \leq \frac{M}{2} \right) \quad (2.2)$$

For binary classification problems where y takes binary values (0 or 1), the following classification rule predicts $y(i)$ based on $\theta(i)$:

$$\hat{y}(i) = \begin{cases} 1 & \theta(i) > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

The naive way of solving this classification problem is to consider each spatiotemporal point of X as one independent feature. However, this is impractical since the total number of features ($N \times M$) can be forbiddingly large. To reduce the number of features, X is projected into a feature space such as in the linear case

$$z_i(n, j) = \sum_{-\frac{M}{2} \leq \tau \leq \frac{M}{2}} b_j(\tau) x^{(n)}(t_i - \tau) \quad (2.4)$$

where b_j is the j th basis of the total J basis functions used for feature extraction, $x^{(n)}$ is the n th neuron of the total N neurons included in analysis. By concatenating $N \times J$ projected features into a feature vector, model inputs (i.e., spatiotemporal features) can be significantly reduced from a vector with a dimensionality of $N \times M$ to a vector with a dimensionality of $N \times J$, since $M \gg J$. Equation 2.2 can be thus rewritten as

$$\theta(i) = \text{Prob}(y(i) | [Z_i^{(1)}, Z_i^{(2)}, \dots, Z_i^{(N)}]), \quad (2.5)$$

where $Z_i^{(n)} = [z_i(n, 1), z_i(n, 2), \dots, z_i(n, J)]$.

Equation 2.5 represents a multi-input, single-output (MISO) decoding model of spikes. It decodes spatiotemporal patterns of spikes X_i into the behavioral label y_i based on spatiotemporal features Z_i extracted from X_i . The spike decoding model proposed in this letter is different from the multi-input, multi-output (MIMO) model, which is essentially a regression model for capturing spike-to-spike transformations (Berger et al., 2012; Song, Chan, et al., 2009). The spike decoding model, on the other hand, is a classification model that takes neural spike trains as inputs to decode behavioral/cognitive variables (e.g., image categories) (see Figure 1).

2.2 Regularized Generalized Linear Model. Equation 2.5 can be solved with a generalized linear model (GLM; McCullagh, 1984), where the output signal y is assumed to follow an exponential family distribution with a mean

μ , which can be calculated as a weighted summation of input features $Z_i^{(n)}$ followed by a static nonlinear link function as

$$\theta(i) \sim \text{Prob}(\mu) = f^{-1} \left(\sum_{n=1}^N w^{(n)} Z_i^{(n)} + w_0 \right), \quad (2.6)$$

where f is the link function and $w^{(n)}$ are the sought modeling coefficients that can be estimated with a maximum-likelihood estimation (MLE) method (White, 1982; Rossi, 2019). Assuming that samples of output signals y are conditionally independent, the likelihood function can be expressed as

$$\mathcal{L}(w) = \prod_{i=1}^I \text{Prob}(y(i) | [Z_i^{(1)}, Z_i^{(2)}, \dots, Z_i^{(N)}]) = \prod_{i=1}^I \theta(i)^{y(i)} (1 - \theta(i))^{1-y(i)}, \quad (2.7)$$

where I is the total number of samples.

The goal of MLE is to maximize $\mathcal{L}(w)$ or minimize the negative log-likelihood $-l(w)$ as

$$-l(w) = \sum_{i=1}^I \{y(i) \log \theta(i) + (1 - y(i)) \log(1 - \theta(i))\}. \quad (2.8)$$

Using a *logit* link function, equation 2.6 becomes

$$\theta = \left[1 + \exp \left(- \sum_{n=1}^N w^{(n)} Z_i^{(n)} - w_0 \right) \right]^{-1}. \quad (2.9)$$

When the number of features is large compared to the number of instances, estimation of w becomes an underdetermined problem; $L1$ regularization can be used to achieve sparse model estimation of w . The minimizer S becomes the summation of $-l(w)$ and a penalty term that is expressed as the $L1$ -norm of w as

$$S(w) = -l(w) + \lambda \left(\sum_{n=1}^N \|w^{(n)}\|_1 \right), \quad (2.10)$$

where λ is the regularization hyperparameter that controls the relative importance between the likelihood and the penalty. A larger value of λ leads to a higher degree of the penalty and sparser model estimations. The

optimal value of λ can be selected using a cross-validation procedure (Kohavi, 1995).

2.3 Multi-Resolution Feature Extraction with B-Spline Basis Functions. In this study, the B-spline basis function (de Boor, 1972; Smith & Schumaker, 1982; Song, Marmarelis, & Berger, 2009; Tu, Song, Breidt, Berger, & Wang, 2012) is used to extract temporal features from spike trains. B-splines are piecewise polynomials with smooth transitions between adjacent pieces at a set of interior knot points. Compared to the typical binning method, B-spline can be used as a generalized histogram method (Hlaváčková-Schindler, Paluš, Vejmelka, & Bhattacharya, 2007) to bin spike trains into firing rates with smoother boundaries. Suppose a polynomial spline of degree d in the range of $[0, M]$ has m interior knot points, which form the sequence $\eta_0 = 0 < \eta_1 < \eta_2 < \dots < \eta_m < \eta_{m+1} = M$. The j th B-spline basis function can be defined in a recursive fashion as

$$b_{j,d}^m(\tau) = \frac{\tau - \eta_j}{\eta_{j+d-1} - \eta_j} b_{j,d-1}^m(\tau) + \frac{\eta_{j+d} - \tau}{\eta_{j+d} - \eta_{j+1}} b_{j+1,d-1}^m(\tau), \quad (2.11)$$

where

$$b_{j,0}^m(\tau) = \begin{cases} 1 & \text{if } \eta_j < \tau < \eta_{j+1} \\ 0 & \text{otherwise} \end{cases}. \quad (2.12)$$

With a fixed degree d and a sequence of m interior knots, the total number of B-spline basis functions J is $m + d + 1$.

In this study, m interior knots are evenly distributed in the decoding window $[-\frac{M}{2}, \frac{M}{2}]$. Therefore, given a set of interior knots, the temporal resolution of the B-spline basis is $M/(m + 1)$. In the following text, we simply refer to m as the “B-spline resolutions.” Given a B-spline resolution, the basis b_j described in equation 2.4 can take the form of the j th B-spline basis function directly. As such, the feature dimension is controlled by the B-spline resolution m . A larger m results in a higher feature dimension and a finer temporal resolution. By adjusting m , B-spline features can range from representing rate code to temporal code with varying temporal resolutions.

For the purpose of including multiple temporal resolutions, a wide range of values of m is included (see Figure 2). Each B-spline resolution corresponds to one classifier H_m and a total number of Q classifiers are included.

2.4 Sparse Classification Function Matrix. For every single-resolution model H_m , model coefficients can be reshaped into a matrix with dimensionality as $N \times J$, with each row containing coefficients of all features of one

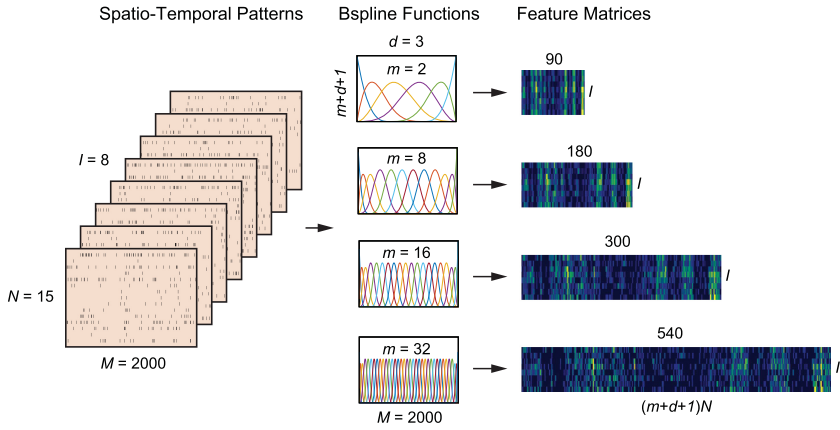


Figure 2: Extracting multi-resolution features from spatiotemporal patterns of spikes using B-spline functions with different resolutions.

neuron. With the coefficient matrix and B-spline basis functions, a sparse classification functional matrix (SCFM) can be calculated as

$$F^m(n, \tau) = \sum_{j=1}^J b_j^m(\tau) w(n, j). \quad (2.13)$$

The SCFM has the same dimension as the spatiotemporal patterns of spikes X . It provides a functional map to relate spike patterns to the decoding target in a probabilistic manner and can be directly used to calculate the conditional probability of the label with the original spatiotemporal patterns X as

$$\begin{aligned} \theta(\hat{y}(i) = 1 | X(t_i - \tau)) \\ = \left[1 + \exp \left(-w_0 - \sum_{n=1}^N \sum_{-\frac{M}{2} \leq \tau \leq \frac{M}{2}} F^m(n, \tau) x^{(n)}(t_i - \tau) \right) \right]^{-1}. \end{aligned} \quad (2.14)$$

Regions of interest (ROIs) refer to nonzero regions of the SCFM where spike patterns have nonzero (positive or negative) contributions to the decoding target y .

For each neuron n , the j th B-spline basis function (modeled with the single-resolution m) can be converted into probabilities with its corresponding model coefficients as

$$B_j^m(\tau) = [1 + \exp(-w_0 - b_j^m(\tau) w(n, j))]^{-1}. \quad (2.15)$$

Equation 2.15 illustrates how each B-spline basis function depends on the estimated model weights and contributes to the conditional probability θ . The baseline probability is estimated as $B_j^{m*} = [1 + \exp(-w_0)]^{-1}$ as $b_j^m(\tau) = 0$. Points on $B_j^m(\tau)$ with a value higher than B_j^{m*} represent regions where a spike in these regions increases the probability of the pattern belonging to a label, that is, $\theta(\hat{y}(i) = 1 | X(t_i - \tau))$; points on $B_j^m(\tau)$ with a value lower than B_j^{m*} represent regions where a spike decreases this probability.

2.5 Bagging-Based Ensemble Classifier. Estimation of model H_m tends to have large variances due to the small sample size. In this study, a bagging-based strategy is used to reduce estimation variances (Breiman, 1996; Efron & Tibshirani, 1997). In this strategy, data are resampled and partitioned into multiple sets of training and validation data for the model estimation and cross-validation. Specifically, we sample the data based on our partition strategy R time to obtain R training and validation data sets. For each data set, a classifier is trained. The optimal hyperparameter is estimated globally by considering all R classifiers; thus, an ensemble estimation with a lower variance is achieved.

We modify the standard partition strategy to handle imbalanced data, where classes of labels have very different sizes that can cause the accuracy paradox (Abma, 2009). Consider a random k -fold partition of a small data set. It is possible that some folds contain only instances of the majority class, where classification becomes impossible. To solve this problem, we sample the two binary labels separately and then combine them to ensure that both training and validation sets contain instances of both classes. A similar partition strategy is used in the stratified k -fold cross-validation (Diamantidis, Karlis, & Giakoumakis, 2000).

2.6 Nested Cross-Validation. Nested cross-validation (Varma & Simon, 2006) is used to avoid overfitting and provide an unbiased evaluation of the model goodness-of-fit (see Figure 3).

The nested cross-validation process consists of two cross-validation loops—outer and inner loops. In the outer loop, the original data set is divided into two subsets: outer training sets and test sets. The test set is used only for measuring model performance. In the inner loop, the outer training set is split into inner training sets and validation sets (see algorithm 1). The inner cross-validation loop is the same as the standard cross-validation. Inner training sets are used to estimate model coefficients, whereas validation sets are used to tune hyperparameters. Optimal hyperparameters are selected within each outer training set across all inner training and validation sets. Finally, the test set in the outer loop is used to assess the model prediction performance.

In this project, we concatenate all predictions from each outer test set and compare it with true labels of output signals as the final model evaluation.

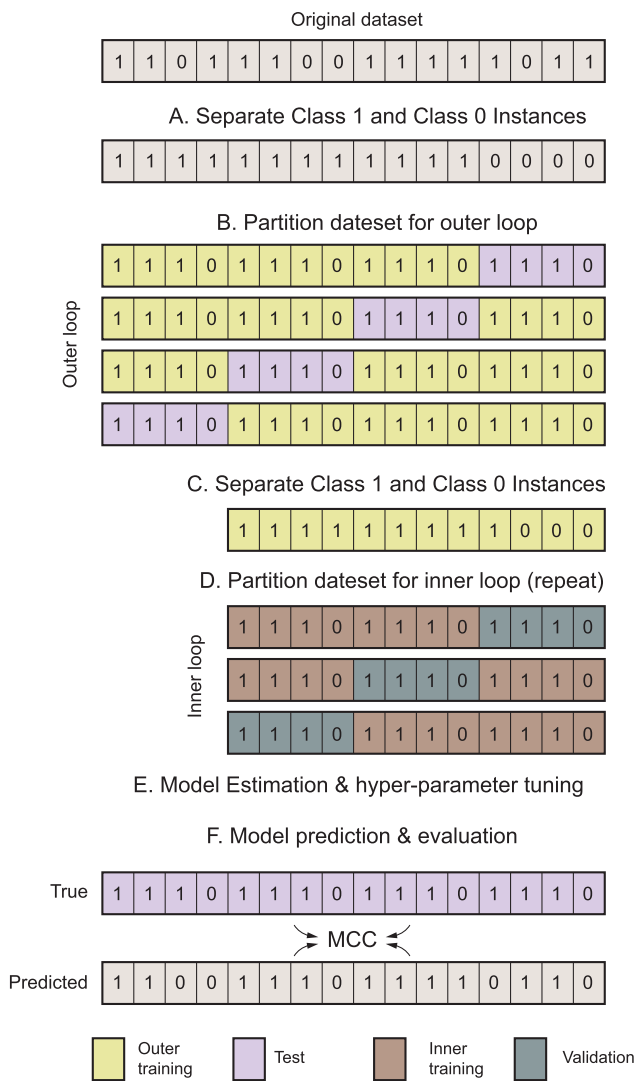


Figure 3: Nested cross-validation procedures are used in model estimation, hyperparameter tuning, and validation.

All steps described above are repeated eight times (i.e., $R = 8$ in algorithm 1) to obtain enough resampled data sets to reduce the estimation variance.

2.7 The Double-Layer Multi-Resolution Spike Decoding Model. As described in section 2.3, multi-resolution models with different B-spline

Algorithm 1: Pseudocode for Estimation of Base-Learner Ensemble Classifiers.

Input: Data set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_I, y_I)\}$;
 Integer R specifying number of models replicating times.
 Percentage P to create sampled training data.

Process:

- % Replicate the modeling processes R times
- 1. for $r = 1, \dots, R$:
- 2. Divide D into two sets D_0 and D_1 based on the output y
 % D_0 contains instances of class 0, and D_1 contains instances of class 1
- 3. Take a sampled replica G_{r0} and G_{r1} by randomly drawing $P = 90\%$ of D_0
 and D_1
 % Combine G_{r0} and G_{r1} into G_r as the training set of this replicate
- 4. $G_r = G_{r0} \cup G_{r1}$
- 5. Take the unsampled points from D_0 and D_1 as G'_{r0} and G'_{r1}
 % Combined G'_{r0} and G'_{r1} into G'_r as the test set of this replicate
- 6. $G'_r = G'_{r0} \cup G'_{r1}$
- 7. end

Output: $\{G_1, G_2, \dots, G_R, G'_1, G'_2, \dots, G'_R\}$

resolutions are trained to include a wide range of temporal resolutions of spikes. However, to solve the classification problem in equation 2.6, one final model is needed to provide the final prediction of output labels. One simple solution is to choose the best model—the one that achieves the highest decoding accuracy—in the ensemble as the final model. However, such a strategy requires intensive computing time and discards all other models except the “best” one. It becomes problematic when the computational resources are limited. More important, such a final model is still a single-resolution model. It is not necessarily true that all neurons share the same temporal resolution. Modeling neurons with a single fixed resolution lacks flexibility and could also harm the decoding performance. The generalization ability of an ensembled multi-resolution model is often much stronger than that of single-resolution models by combining a wide range of resolutions.

In this study, we use a stacking procedure to combine all single-layer single-resolution models together to form a double-layer, multi-resolution ensemble model (Breiman, 1996; Smyth & Wolpert, 1998; Wolpert, 1992). Through this approach, multiple single-resolution models serve as the first-layer “base learners.” In this layer, all base learners classify the input and make their output predictions in the form of probabilities. In the second

Algorithm 2: Pseudocode for Estimation of the Meta-Learner Ensemble Classifier.

Input: $D = \{G_1, G_2, \dots, G_R, G'_1, G'_2, \dots, G'_R\}$; % Data set after bagging
 First layer models H_1, H_2, \dots, H_Q % Q is the number of base learners
 The second layer stacking model H'

Process:

1. for $m = 1, \dots, Q$: % All resolutions of the B-spline basis function
2. $L_m = H_m(G_1, G_2, \dots, G_R)$ % Train first layer base learner L_m given the resolution m with training sets
3. end
4. $D' = \emptyset$ % Create a new data set for storing first layer outputs
5. for $m = 1, \dots, Q$: % All base learners
6. $\theta_m = L_m(G_1, G_2, \dots, G_R)$ % Use L_m to make predictions
7. end
8. $D' = D' \cup \{(\theta_1\theta_2, \dots, \theta_Q)\}$ % Formulate the new data set with all first layer outputs
9. $L' = H'(D')$ % Train the second layer meta-learner with new data set D'
10. $\theta' = L'(G'_1, G'_2, \dots, G'_R)$ % Test the meta-learner with the test set

Output: θ'

Note: Steps 2 and 6 also use bagging and nested cross-validation.

layer, all first-layer output predictions are taken as inputs of a classifier, termed the “meta-learner,” to produce the final output prediction. The pseudocode of our double-layer, multi-resolution classification model is shown in algorithm 2.

By applying the same sigmoid classification rule described in equation 2.3, we can get the final ensembled binary classification of the output label as

$$\hat{y}(i) = \begin{cases} 1 & \text{if } \theta'(i) > 0.5 \\ 0 & \text{otherwise} \end{cases}. \quad (2.16)$$

With stacking, the second-layer model combined all first-layer models with their corresponding second-layer model coefficients w' , which has a dimension of $1 \times Q$. In addition, an ensemble SCFM can be calculated from first-layer SCFMs and their corresponding second-layer coefficients as

$$F'(n, \tau) = \left[1 + \exp \left(-w'_0 - \sum_{m=1}^Q F^m(n, \tau) w'(m) \right) \right]^{-1}. \quad (2.17)$$

Similarly, B-spline basis functions in units of probability can also be calculated with the second layer as

$$B'_j(\tau) = \left[1 + \exp \left(-w'_0 - \sum_{k \neq m}^Q w'(k) B_j^{k*} - B_j^m(\tau) w'(m) \right) \right]^{-1}. \quad (2.18)$$

The baseline probability of such double-layer probability curves is defined by baselines from all first-layer B_j^{m*} . Equations 2.17 and 2.18 show how the double-layer SCFM combines a wide range of resolutions and contributes to the conditional probability θ .

Now we have a complete double-layer, multi-resolution classification model for decoding spatiotemporal patterns of spikes. In the first layer, multi-resolution, B-spline basis functions are used to explore neural temporal resolutions and reduce data dimensionalities. Multiple LASSO classifiers are trained as first-layer base learners to classify the spatiotemporal patterns of spikes based on the B-spline features. Inputs of those base learners are B-spline features with different resolutions. Outputs are behavioral labels in memory-dependent tasks (see section 2.9). In the second layer, another LASSO classifier is used to combine all single-resolution base learners to form a final multi-resolution meta-learner. As such, inputs of the meta-learner are probabilities estimated by base learners, while outputs are corresponding behavioral labels (see Figure 4).

2.8 Model Evaluation. Modeling performance is evaluated with the Matthews correlation coefficients (MCCs), which is robust to imbalanced data. MCC is calculated from the numbers of true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*) in the confusion matrix as

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}. \quad (2.19)$$

MCC values of 1, 0, and -1 indicate perfect, random, and opposite predictions to output labels, respectively. Note that trivial models, which predict outputs all as 1 or all as 0, also yield an MCC of 0.

2.9 Experimental Procedures. Two types of experimental data are used to test the double-layer decoding model in this study. One is hippocampal spiking activities recorded from rats performing a spatial delayed nonmatch-to-sample (DNMS) task (Deadwyler et al., 1996). The other is hippocampal spiking activities recorded from human epilepsy patients performing a visual delayed match-to-sample (DMS) task (Hampson et al., 2018).

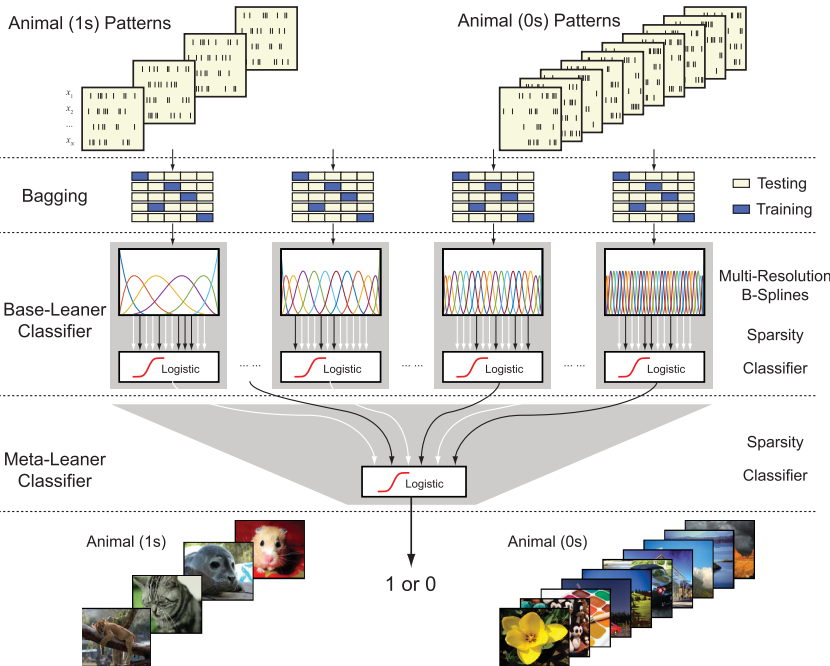


Figure 4: Double-layer, multi-resolution classification model of spikes. First row: Spatiotemporal patterns of spikes. Second row: Bagging-based partition strategy. Third row: First-layer single-resolution base learners in the form of sparse logistic B-spline classifiers. Fourth row: Second-layer multi-resolution meta-learner in the form of a sparse logistic classifier. Fifth row: Model outputs in the form of binary signals.

2.9.1 *Delayed Nonmatch-to-Sample Task in Rats.* Hippocampal CA3 and CA1 spike trains were recorded from rats ($n = 5$) performing the DNMS task using multi-electrode arrays (MEAs). Each DNMS trial started with the sample phase (see Figure 5, top) when the animal was presented randomly with one of the two levers (left or right) on one wall of the chamber. The animal was trained to press the lever to generate a sample response event. The lever was then retracted, and the delay phase with a random duration (5–30 sec) started. For the delay duration, the animal was required to nose-poke into a lighted device on the opposite wall. When the delay ended, the nose-poke light was turned off; both levers were extended, and the animal was required to press the lever opposite the one it pressed during the sample phase and generated a nonmatch response. If the correct lever was pressed, the animal was rewarded with a drop of water. One session had 80 to 120 DNMS trials. Spatiotemporal patterns of CA3 and CA1 spikes around

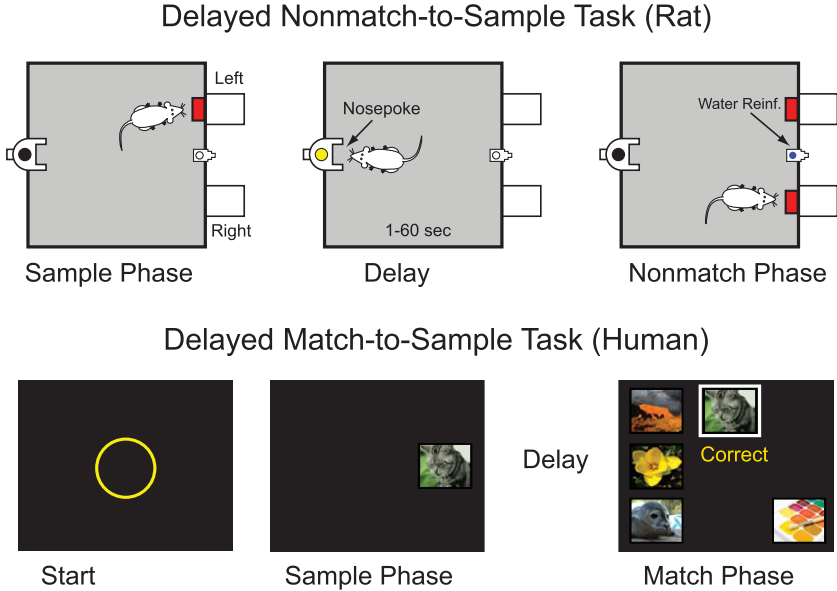


Figure 5: Memory-dependent behavioral tasks used in this study. Top: Spatial delayed nonmatch-to-sample task in rodent studies. Bottom: Visual delayed match-to-sample task in human studies.

(−2 sec to 2 sec) sample response events were used as model inputs. Lever locations (left: 1; right: 0) were used as model outputs.

2.9.2 Delayed Match-to-Sample Task in Human Patients. Hippocampal CA3 and CA1 spike trains were recorded from human epilepsy patients ($n = 2$) performing a DMS task using Ad-Tech “macro-micro” depth electrodes (Hampson, Pons, Stanford, & Deadwyler, 2004; Hampson et al., 2018) (Figure 5, bottom). Each DMS trial was initiated with a red focus ring presented on the touch screen. The subject was asked to click on the focus ring to start the sample phase. During the sample phase, one image was presented on the touch screen. The subject was asked to remember the image and then press on the image to form a sample response event. After pressing, the image disappeared, and the delay phase (3–5 sec) started. When the delay phase finished, multiple images including the image presented at sample phase (sample image) were presented on the touch screen. The subject should choose and press on the sample image to generate a correct match response. One session consisted of 100 to 150 DMS trials. Spatiotemporal patterns of CA3 and CA1 spikes around −2 sec to 2 sec sample response events were used as model inputs. The categories of sample images were used as model outputs. Five categories represented with five memory

labels—Animal, Building, Plant, Tool, and Vehicle—were used as model outputs in this study.

All procedures were reviewed and approved by the Institutional Review Boards of Wake Forest University and the University of Southern California, in accordance with the National Institutes of Health. The subjects provided voluntary written informed consent prior to participation in this study. All human experiments were performed at the Comprehensive Epilepsy Center of Wake Forest Baptist Medical Center, Keck School of Medicine of the University of Southern California, and Rancho Los Amigos National Rehabilitation Center.

2.10 Source Code Sharing. The Matlab source code of the memory decoding model is available at <https://github.com/neural-modeling-and-interface-lab/Double-Layer-Multi-Resolution-Memory-Decoding-Model>.

3 Results

3.1 Simulation Studies. First, we tested the proposed double-layer, multi-resolution decoding model with synthetic data to see whether it can decode spatiotemporal patterns of spikes generated with various temporal resolutions. Synthetic spike patterns were simulated with different forms of probability intensity functions (PIFs), which consist of a baseline firing rate f_0 (5 Hz) and gaussian peaks that can be described as

$$P_f(t) = P_{f_0}(t) + \gamma \times \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(t-\mu)^2}{2\sigma^2}}, \quad (3.1)$$

where P_{f_0} is the baseline firing probability, γ is the peak intensity, μ is the center, and σ is the standard deviation of the gaussian function that controls the temporal resolution of simulated spike patterns.

We designed four simulations with progressively higher complexity to examine how the proposed model performs with (1) a single neuron with a wide (low-resolution) peak, (2) a single neuron with a narrow (high-resolution) peak, (3) two neurons with wide and narrow peaks respectively, and (4) population of neurons with randomly generated peaks. Simulation parameters of PIFs are shown in Table 2.

In the first three simulations, two classes (class 1 and class 0) of patterns were simulated. For class 1, PIFs were designed as in equation 3.1 and Table 2; for class 0, PIFs were designed to be flat lines (i.e., homogeneous Poisson spike trains) with values equal to the mean PIF values of their corresponding class 1 patterns. Therefore, the mean firing rates of both classes were the same. Each class was stimulated with 100 instances. Every instance was 4 seconds long. In the fourth population simulation, multiple classes of patterns were simulated with PIF parameters randomly generated in the ranges given in Table 2. In other words, each neuron exhibited

Table 2: PIF Parameters Used in Simulation Studies.

	μ	σ	γ
Single (low)	1	0.3	0.02
Single (high)	3	0.005	0.002
Combined two	1, 3	0.3, 0.005	0.02, 0.002
Population	0.1–3.9	0.001–0.1	0.002–0.02

different PIF patterns corresponding to multiple classes. In this study, multiple classes were classified with multiple binary classifiers, where in each classifier, the class to be classified was treated as class 1, while all other classes were treated as class 0.

In all classifications, bagging replicates R is 8; B-spline resolution m was selected from the range $[0:25, 50:5:150]$. Thus, 47 base learners H_m were trained in the first layer. $[a:b:c]$ denotes the vector of possible values from a to c with step size b . All models were built through Matlab 2019a.

3.1.1 Decoding Single-Neuron Patterns. We started with the simplest two cases, where class 1 patterns had PIFs with low- and high-resolution gaussian peaks, respectively (see Figure 6A). PIFs of class 0 are flat lines and not shown for simplicity. It is worth noting that since the average firing rates of class 1 patterns and class 0 patterns are designed to be the same, these two patterns cannot be classified by any firing rate-based classifier. The main goal here is to gain insight into how the double-layer classifier captures different temporal resolutions in the patterns with different B-splines. B-splines selected by the double-layer model and ensemble SCFMs in the unit of probability, described in section 2.4, were calculated to show the ROIs and compared with true PIFs (see Figure 6B, red lines) and effective PIFs (see Figure 6B, blue lines). True PIFs refer to the analytical PIFs defined in equation 3.1, while effective PIFs are spike histograms calculated after simulations. These two types of PIFs converge when the number of simulated instances is large.

Decoding results show that ensemble SCFMs (see Figure 6B, shaded areas) well match the PIFs used in simulations (see Figure 6A, red and blue lines). As expected, low-resolution B-splines (smaller m) and high-resolution B-splines (larger m) are selected under the wider and narrower gaussian peaks, respectively, to construct the corresponding SCFMs. In both low- and high-resolution cases, the model also selects a few high-resolution B-splines out of the gaussian peaks of true PIFs (see Figure 6A, red lines) due to the random fluctuation in effective PIFs caused by small sample sizes and low firing rates.

We further plot MCCs of all base learners and the meta-learner (see Figure 6C). In the low-resolution case, MCCs of single-resolution base

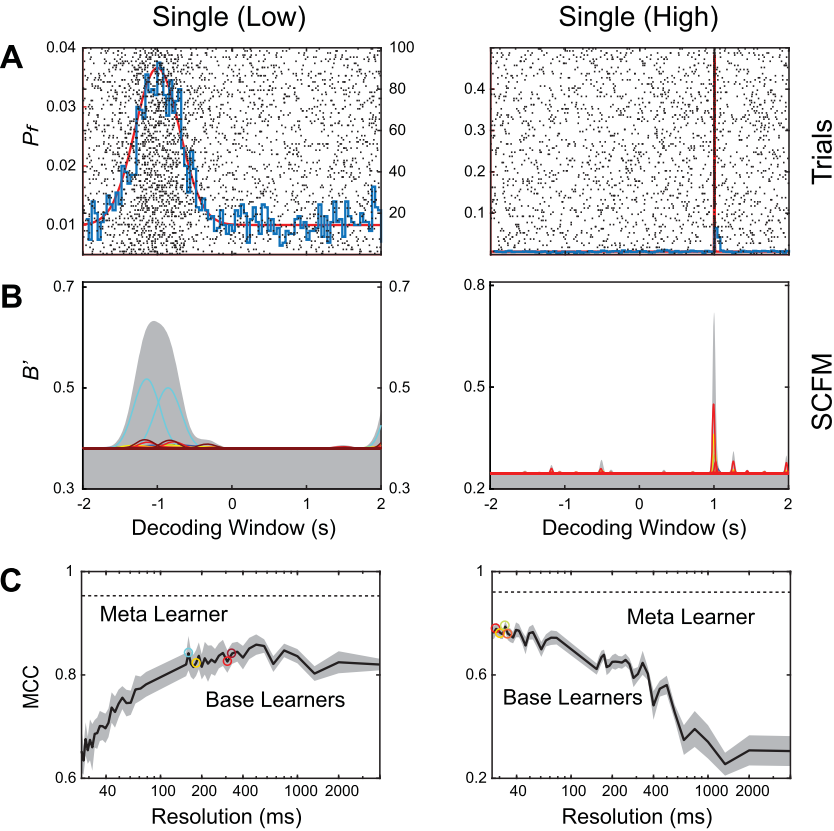


Figure 6: Decoding single-neuron patterns. Left column: Decoding patterns with a low-resolution gaussian peak. Right column: Decoding patterns with a high-resolution gaussian peak. (A) Spike raster plots (black dots), true PIFs (red lines), and effective PIFs (blue lines) of simulated spike patterns. Note that only class 1 patterns are shown. (B) B-splines (color lines) selected and SCFM (shaded areas) estimated by the double-layer model. (C) MCCs of base learners (black lines: mean; shaded area: STD) and meta-learners (dashed lines). Colored circles represent base learners selected by the meta-learner.

learners (see Figure 6C, solid line) reach maximal value at the optimal temporal resolution of B-splines ($m = 7$). In the high-resolution case, one of the maximal temporal resolutions ($m = 120$) yields the highest MCC (see Figure 6C, solid line). In both cases, the multi-resolution meta-learners (see Figure 6C, dashed lines) further boost the MCCs (0.922 in Figure 6C, left, and 0.871 in Figure 6C, right) on top of the highest MCCs (0.844 in Figure 6C, left, and 0.754 in Figure, 6C right) achieved by the best single-resolution base learner.

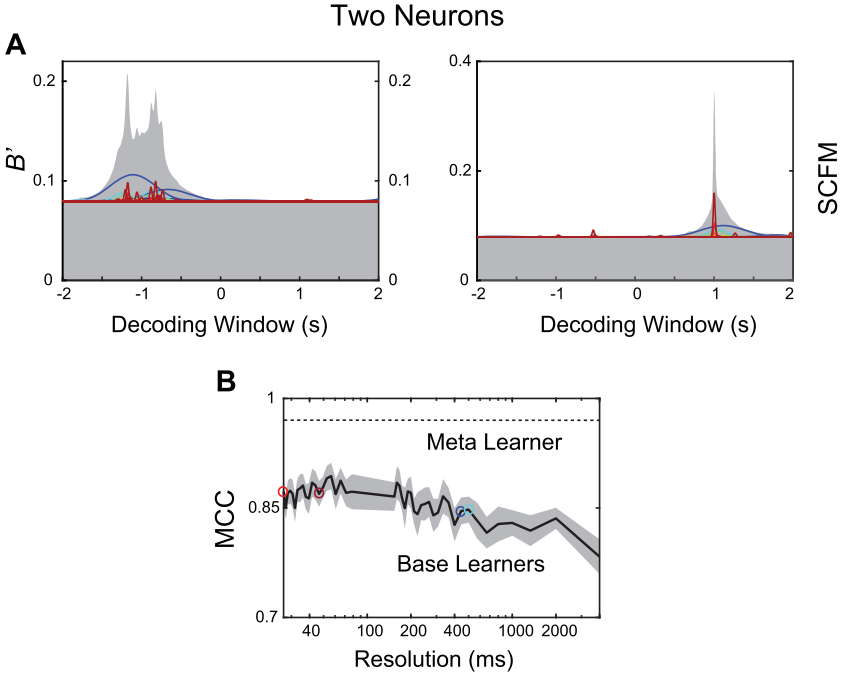


Figure 7: Decoding two-neuron patterns. (A) B-splines (color lines) selected and SCFM (shaded areas) estimated by the double-layer model. (B) MCCs of base learners (black lines: mean; shaded area: STD) and meta-learners (dashed lines). Colored circles represent base learners selected by the meta-learner.

It is worth noting that there is no multi-resolution PIF in such single-neuron simulations. However, the meta-learner can still boost the performance significantly by combining base learners with different B-spline basis functions since each individual base learner is imperfect in capturing the PIF due to its specific B-spline knot sequence (see Figure 6C, colored circles).

3.1.2 Decoding Two-Neuron Patterns. In the second simulation, we combined the two neurons above to form two-neuron patterns for decoding. The goal is to see how the double-layer model decodes patterns with multiple temporal resolutions—wide, and narrow gaussian peaks, with its base learners and the meta-learner.

Decoding results show that the ensemble SCFM (concatenation of SCFM in Figure 7A) well matches the PIFs used in simulations (see Figure 6A, red and blue lines). Both low- and high-resolution B-splines under gaussian peaks are selected by the meta-learner and used to construct the SCFM. Among the four groups of B-splines (represented by four different

colors) selected by the meta-learner, two are high resolutions (see Figure 7A, dark and light red lines), and two are low resolutions (see Figure 7A, dark and light blue lines).

These four groups of B-splines correspond to four base learners with different temporal resolutions. Similar to single-neuron pattern decoding results, low- and high-resolution B-splines show relatively larger contributions to the wider and narrower gaussian peaks in the PIFs, respectively. Different from single-neuron results, the meta-learner here tends to use a mixture of low- and high-resolution B-splines to fit both wide and narrow gaussian peaks. This is because each base learner uses a single temporal resolution to fit both wide and narrow peaks. Although each base learner is imperfect in fitting all peaks, the meta-learner combined multiple base learners to achieve good fits to the PIFs (see Figure 7A). It is worth noting that with both low- and high-resolution B-splines, the SCFM captures the noisier effective PIF instead of the smooth, true PIF. Therefore, the “noisy” SCFMs in Figure 7A are caused not by overfitting but by the small sample size, which biases the true PIF.

MCCs of all base learners and the meta-learner are shown in Figure 7B. Results show that with two neurons, base learners in general perform better than with single neurons due to the redundant information in the two-neuron patterns (i.e., base learners can achieve relatively higher MCCs from classifications in either neuron). More important, the meta-learner selects both low- and high-resolution base learners to classify the two-neuron patterns that contain both wide and narrow gaussian peaks (see Figure 7B, colored circles). By combining multiple resolutions, the meta-learner boosts the MCC to 0.95, which is much higher than the MCC achieved by the best base learner (0.89).

3.1.3 Decoding Population Patterns. Our last simulation dealt with population-level spatiotemporal patterns of spikes with a larger number of neurons and more complex temporal patterns. The population patterns were simulated with 30 neurons and 5 categories. These numbers were chosen based on the typical numbers we encountered in human experiments. Each neuron had a 0.5 probability of having no gaussian peaks and a 0.25 probability of having one or two peaks, respectively. The gaussian peaks have randomly chosen locations, widths, and intensities (see Table 2 and Figures 8A and 8B). For each category, 500 instances were simulated.

Five double-layer, multi-resolution decoding models were built for the five categories (see Figure 8). Results show that these models can perfectly decode the population patterns ($\text{MCC} = 1$ in all categories). The resulting SCFMs well capture most ROIs in PIFs (see Figures 8A and 8C). As proved in equation 2.14 and shown here, SCFMs provide characteristic spatiotemporal maps for decoding spatiotemporal patterns by identifying ROIs unique for each class (category). In other words, SCFMs are essentially the spatiotemporal areas that maximize the difference between class 1

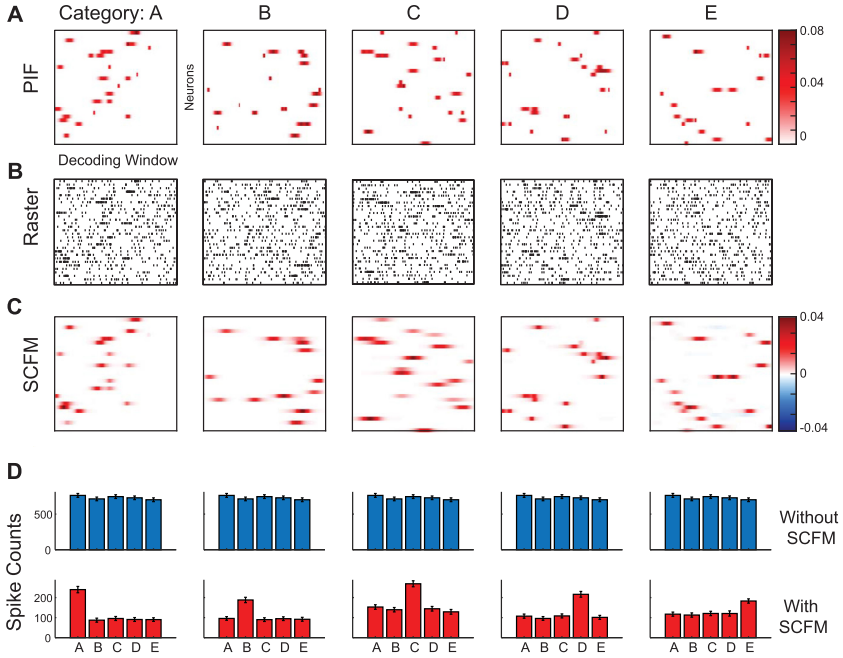


Figure 8: Decoding population patterns. (A) PIFs of the five categories. (B) Raster plots of one example of spatiotemporal patterns of spikes simulated based on PIFs. (C) SCFM of double-layer decoding models. In panels A, B, and C, the x -axis is the decoding time window (-2 sec to 2 sec), and the y -axis is indices of neurons. PIF, raster and SCFM all have the same dimensionality. (D) Spike counts with and without SCFMs as masks. Bars: Mean spike counts per instance. Error bars: STD of spike counts.

patterns and class 0 patterns for classification. This is clearly shown in Figure 8D: without applying SCFMs (i.e., counting all spikes in the spatiotemporal patterns), there is no significant difference between spike counts in the patterns across different categories (see Figure 8D, top panel); with SCFMs applied (i.e., counting spikes within nonzero areas of SCFMs only), the difference between spike counts across different categories becomes obvious (see Figure 8D, bottom panel). Note that this simple counting procedure is not used in the decoding model: such results are illustrated to intuitively demonstrate how the SCFMs identify ROIs. The actual model calculation is shown in equations 2.6, 2.14, and 2.17.

3.1.4 Factors Influencing Classification Performance. At the end of the simulation study, we investigated the influence of key experimental parameters

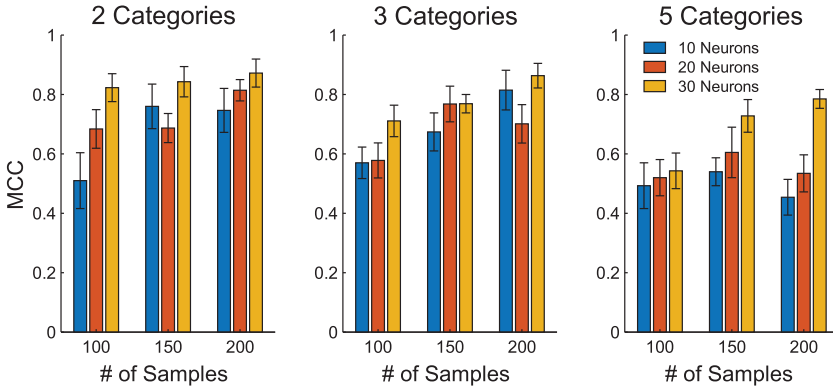


Figure 9: Influence of key experimental parameters on classification performance. Bars: Mean MCCs. Error bars: STD of MCCs.

on the model performance. These parameters include the number of neurons N , the number of instances per category T , and the number of categories C . Since 500 instances per category result in perfect classification as shown in the previous section, we decrease T to smaller numbers that are more realistic in experimental or clinical settings. Twenty-seven simulations and classification were performed with a different combination of N (10, 20, 30), T (100, 150, 200), and C (2, 3, 5) (see Figure 9).

Results show that as expected, decoding performance increases with the number of instances. Second, including more neurons increases model performance. This result indicates that the decoding model does not suffer from overfitting caused by including a larger number of features from more neurons. Finally, having more categories decreases model performance as data become more imbalanced. However, even in the most challenging case with the fewest instances (100), the fewest neurons (10), and the most categories (5), the decoding model can still achieve a significant level of classification ($\text{MCC} = 0.493 \pm 0.077$).

3.2 Decoding with Experimental Data. We tested the double-layer decoding model with hippocampal spiking data recorded from rodents and human epilepsy patients performing memory-dependent behavioral tasks.

3.2.1 Decoding Binary Spatial Locations from Hippocampal Spikes in Rats. In rat decoding models, model inputs are spatiotemporal patterns of spikes recorded from hippocampal CA3 and CA1 neurons around sample response events (-2 sec to 2 sec). Model outputs are locations of the lever (left or right) pressed at sample response events represented as a binary variable (1 or 0). We built both single-layer and double-layer decoding models using data from five animals. Results show that both can accurately decode

Table 3: Model Performances of Single-Layer and Double-Layer Decoding Models in Rodent Data.

Rat	Number of Neurons	Number of Trials	Single Layer (MCC)	Double Layer (MCC)
a	50	89	0.952 ± 0.008	0.977 ± 0.02
b	26	63	0.835 ± 0.038	0.943 ± 0.044
c	57	53	0.929 ± 0.032	0.989 ± 0.018
d	41	68	0.956 ± 0.015	0.964 ± 0.022
e	56	68	0.826 ± 0.031	0.945 ± 0.031

the sample locations (see Table 3). In addition, double-layer models consistently outperform their corresponding single-layer models with significantly higher MCCs (paired t -test, $p < 0.001$ in all cases).

Figure 10 shows examples of decoding models from two animals. Since PIFs are unknown in experimental data, peri-event histograms of neurons around sample responses are shown instead. There is no obvious difference between the left and right histograms in both animals (see Figure 10A). This is also indicated in Figure 10D, which shows little difference in spike counts between left and right without applying the SCFM mask. However, with SCFM applied, the difference in spike counts between the two patterns becomes much more apparent. In both models, SCFMs exhibit both positive and negative ROIs.

3.2.2 Decoding Visual Memory Categories from Hippocampal Spike in Humans. In human decoding models, model inputs are spatiotemporal patterns of spikes recorded from hippocampal CA3 and CA1 neurons around sample response events (-2 sec to 2 sec) of the DMS task. Patients 1 and 2 performed 145 and 109 trials and had 55 and 80 neurons recorded, respectively. Model outputs are binary labels (1 or 0) corresponding to the five categories of sample images (see Table 4). Since these five categories are not mutually exclusive, we formulate the decoding as a multi-label classification problem (as opposed to a multi-class classification problem), in which each label takes binary values. Independent single-layer and double-layer decoding models are built for each category for each patient.

Results show that both single-layer and double-layer decoding models can classify the spike patterns with respect to the image categories. They both achieve significantly higher-than-zero MCC values in all categories in both patients. In addition, double-layer models perform significantly better than their corresponding single-layer models (paired t -test, $p < 0.001$ in all cases).

Details of two human decoding models are shown in Figures 11 and 12. Similar to what is shown in rodent models, there is no obvious difference

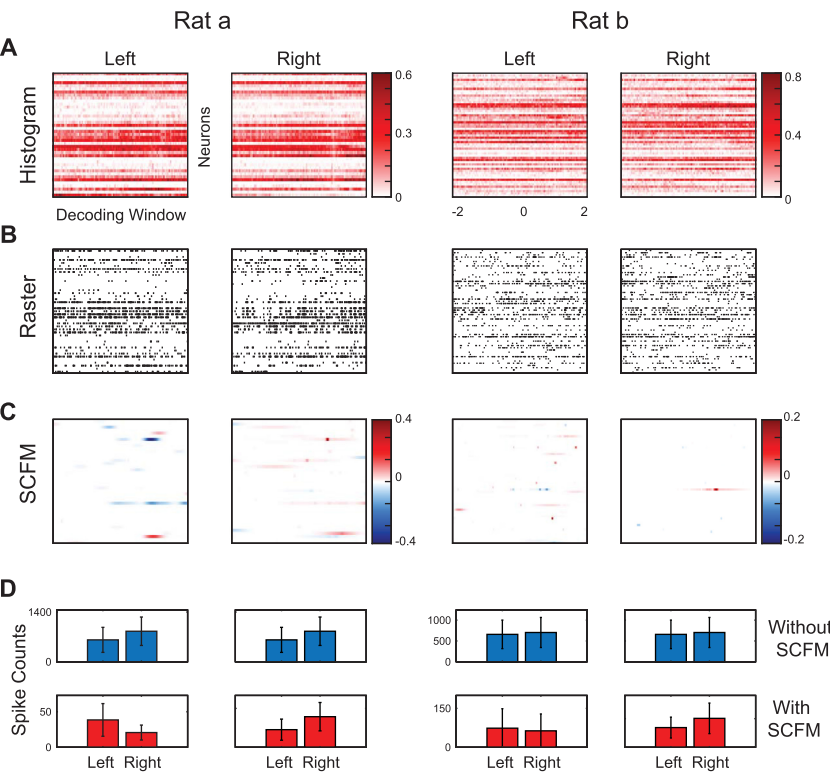


Figure 10: Decoding hippocampal spike patterns in rats performing a DNMS task. Left: rat a. Right: rat b. (A) Peri-event histograms of spike patterns during left and right sample responses. (B) Raster plot of one spatiotemporal patterns of spikes during left and right sample responses. (C) SCFMs of double-layer decoding models. (D) Spike counts without and with SCFMs as masks.

in the peri-event histograms of spikes across different categories (see Figure 11A). However, the decoding models identify the sparsely distributed spatiotemporal regions of spike patterns that best differentiate the patterns across different categories (see Figure 11D). These regions are visualized with the SCFMs, which are characteristic of each category.

3.3 Contribution of the Bagging Procedure. Bagging procedure is included in our modeling methodology to facilitate model estimation and reduce estimation variances caused by the small sample size. To demonstrate the contribution of the bagging procedure, we built decoding models with and without bagging and compared their MCC performances

Table 4: Model Performances of Single-Layer and Double-Layer Decoding Models in Human Data.

Patient	Category	MCC (Single Layer)	<i>p</i> -values	MCC (Double Layer)	<i>p</i> -values
1	Animal	0.242 ± 0.093	0.002	0.364 ± 0.101	<0.001
	Building	0.486 ± 0.068	<0.001	0.624 ± 0.046	<0.001
	Plant	0.195 ± 0.061	<0.001	0.245 ± 0.101	<0.001
	Tool	0.198 ± 0.116	0.004	0.309 ± 0.108	<0.001
	Vehicle	0.218 ± 0.122	0.007	0.432 ± 0.106	<0.001
2	Animal	0.213 ± 0.036	0.007	0.253 ± 0.168	<0.001
	Building	0.314 ± 0.036	<0.001	0.462 ± 0.070	<0.001
	Plant	0.238 ± 0.037	<0.001	0.242 ± 0.088	<0.001
	Tool	0.392 ± 0.002	0.016	0.479 ± 0.070	0.002
	Vehicle	0.416 ± 0.017	0.002	0.469 ± 0.093	<0.001

(see Figure 13). In the models without bagging, hyperparameters are optimized within each inner loop and then used in outer loop testing set.

Results show that the decoding models without the bagging procedure underperform their corresponding decoding models with bagging in all synthetic and experimental data sets. Specifically, models with bagging consistently yield MCCs with higher mean values and lower STDs.

4 Discussion and Conclusion

This letter presents a double-layer, multi-resolution decoding model for decoding behavioral and cognitive information from spatiotemporal patterns of spikes. This model structure is designed specifically for solving the highly underdetermined estimation problem caused by high-dimensional spike patterns, the small sample size, and taking into account the unknown optimal temporal resolution for decoding spikes. In first-layer base learners, B-spline basis functions with a single temporal resolution are used to extract spatiotemporal features from spike patterns and reduce dimensionality. $L1$ -regularized logistic classifiers are used to yield sparse estimations of model coefficients as solutions to the underdetermined problems (Berens et al., 2012). Estimating multiple copies of classifiers using resampled data allows us to build ensemble classifiers to reduce estimation variance and thus obtain more stable model estimation. With the second-layer meta-learner, multiple base learners are fused in a data-driven manner to incorporate multiple temporal resolutions into the model and render the decoding model double layer and multi-resolution. To avoid overfitting, nested cross-validation is used throughout estimations. Results in both synthetic and experimental data show that this model can accurately decode high-dimensional spike patterns using a small number of instances. The double-layer model consistently outperforms the single-layer model by

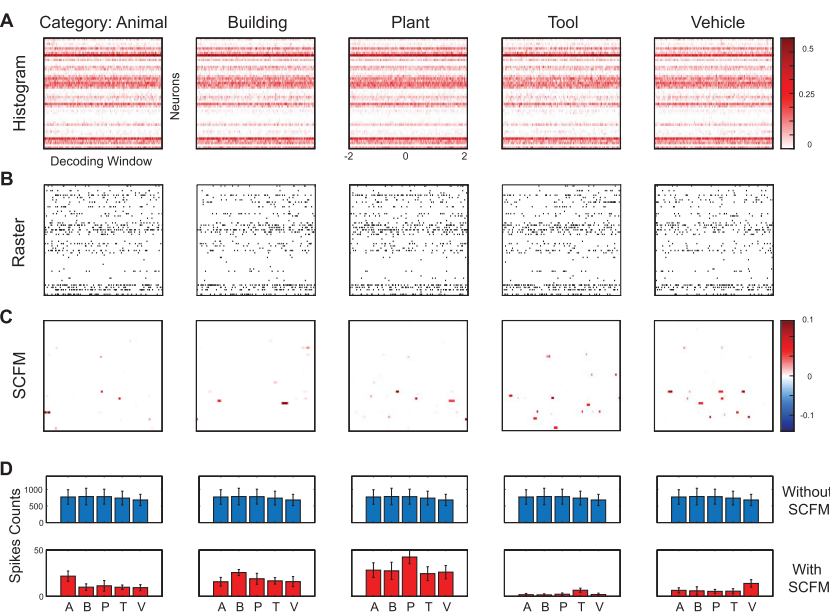


Figure 11: Decoding hippocampal spike patterns in human performing a DMS task (subject 1). (A) peri-event histograms of spike patterns during sample responses. (B) Raster plot of one spatiotemporal patterns of spikes during sample responses. (C) SCFMs of double-layer decoding models. (D) Spike counts with and without SCFMs as masks.

utilizing multi-resolution spatiotemporal features of spike patterns in classification. The decoding models also provide signature functions (i.e., SCFMs) that represent spatiotemporal characteristics of spike patterns most relevant to classifications.

In the rodent study, the decoding model yields almost perfect model predictions. In the human study, the decoding performances are all significantly above chance level but lower than those in the rodent study. The possible reasons for this difference include (1) a higher level of noise in human recordings (Hampson et al., 2018) due to different recording techniques and procedures; (2) more imbalanced data in the human decoding model (five labels representing five memory image categories) than in the rodent decoding model (single label representing left or right locations of the sample lever); and (3) differences between spatial encoding and image (object) encoding, that is, there could be more cells encoding spatial information (place cells) than object information in the hippocampus.

The double-layer decoding model described here bears some resemblance to an artificial neural network (ANN). For example, both are supervised learning models with a layered structure. The logistic function is also

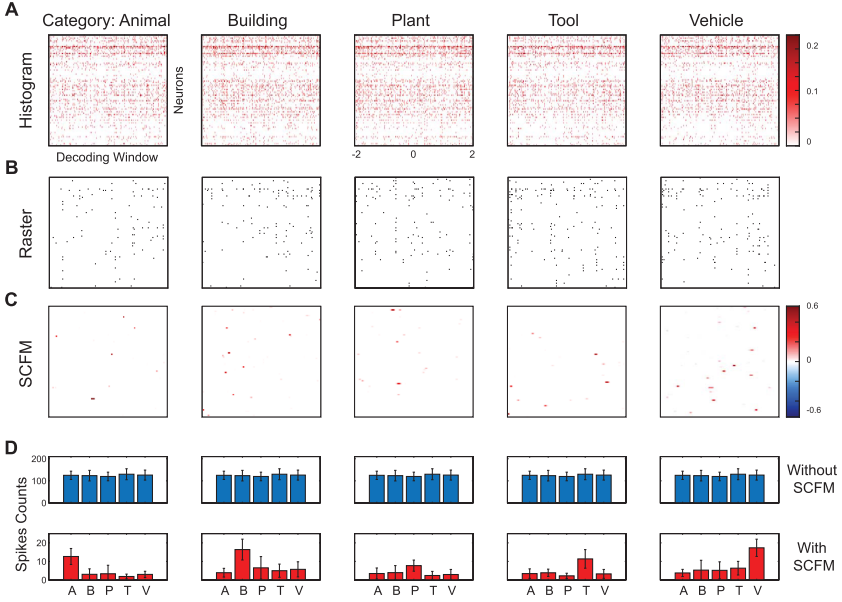


Figure 12: Decoding hippocampal spike patterns in human performing a DMS task (subject 2). (A) Peri-event histograms of spike patterns during sample responses. (B) Raster plot of one spatiotemporal patterns of spikes during sample responses. (C) SCFMs of double-layer decoding models. (D) Spike counts with and without SCFMs as masks.

a commonly used activation function in ANN. However, fundamental differences between these two models exist in that ANN originated from a connectionism learning theory where simultaneous processing of distributed computational units is used. In contrast, the double-layer decoding model described here is inspired by functional expansion and ensemble learning theories, where multiple independently estimated computational units emphasizing different aspects (e.g., temporal resolutions) of the modeled system are combined to form the final model. The main reason behind this choice is the highly underdetermined estimation problem caused by the high dimensionality of input signals and, more important, the very short data length (100–200 instances). Given such data length, model estimations in every step can include only a minimal number of open parameters to make model estimations possible and avoid overfittings. In our double-layer decoding model, every base learner is estimated with all (training) data points available. The meta-learner again uses all (training) data points to determine the weights of each base learner. This somewhat “greedy” training strategy is necessary for the highly underdetermined problems being solved in this study but is drastically different from what is typically

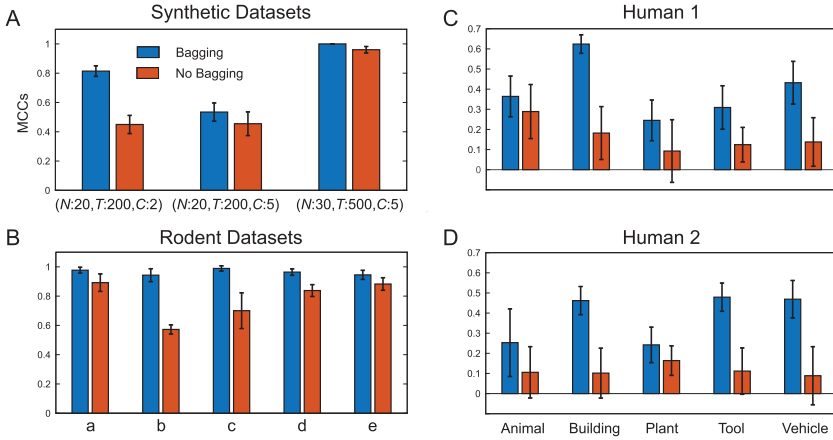


Figure 13: Comparisons of decoding performances between the proposed model and the model without the bagging procedure. (A) Synthetic data sets. (B) Rodent data sets. (C) Human data set 1. (D) Human data set 2. Blue bars: Mean MCCs of the proposed model. Orange bars: Mean MCCs of the model without bagging. Error bars: STD.

used in ANNs (e.g., backpropagation). When the standard backpropagation method is used, all model parameters of the ANN are simultaneously trained based on the error signal. This will cause serious overfitting or non-convergence when the sample size is small, as in this study. The ensemble learning approach greedily estimates model parameters group by group and step by step to solve the underdetermined estimation problem and mitigate overfitting. It can be considered a very special way of training an ANN with small sample size.

Modeling factors involved in this study can be further explored to improve the model performance in future studies. The current model is computationally intensive and requires tens to hundreds of hours on a single personal computer. Speeding up model estimation is critical for experimental and clinical applications (She, Robinson, Berger, & Song, 2020). For example, the $L1$ regularization rate (i.e., λ in equation 2.10), which controls the sparsity level, is a key determinant of the total computing time since model estimations need to be repeated for each λ value. In our current strategy, λ is treated as a hyperparameter; its optimal value is searched exhaustively within a certain range (10^{-5} to 10^0) with a high resolution using the nested cross-validation procedure. The sequence of candidate λ values can potentially be reduced and optimized based on empirical results to speed up model estimations. Similarly, the number of temporal resolutions to be explored can also be optimized to include only the more possible temporal resolutions. As the starting step, we use overcomplete sets of

hyperparameters in this study to impose minimal constraints on the models; as more insights are gained from more analyses using experimental data, the hyperparameter space is expected to be greatly reduced without sacrificing the model's performance.

As shown in the results, the double-layer decoding model drastically improves the model performance with its ensemble learning components (i.e., bagging and stacking). This strong improvement results from the ensemble learning methods and is also due to the small sample size used in this study. In the data used in this study, particularly the experimental data, the sample sizes are so small that most of the standard estimation methods fail with no above-chance performances; even with regularized estimation, most single-resolution, single-trial (no bagging) base learners have moderate performances. This leaves enough room for the ensemble learning methods to boost the model performance by combining multiple weak learners to form a strong learner. In contrast, when the sample size is large, the improvement caused by ensemble learning methods is expected to be less prominent (Glaser et al., 2020).

In the current version of the double-layer, multi-resolution decoding model, output labels are treated independently; a binary classification model is built for each output label using only input (spike patterns) features. The dependencies between output labels are not included. For example, the human spike decoding models are five independent decoding models for five memory (sample image) categories. This is mainly because the sample images are hand-selected to be relatively independent (Song et al., 2016, 2017). The dependence between image categories thus is weak and unrealistic to be generalized to model other images. In extension to larger-scale decoding models with a larger number of categories and naturalistic distribution of sample images, the decoding model can be modified to explicitly incorporate dependencies between image categories into the model using a graphical modeling approach (Lanchantin, Sekhon, & Qi, 2020; Zha et al., 2008; Zhou et al., 2018; Zhou & Zhang, 2007). By exploiting such dependencies, model decoding performances are expected to be improved further.

Acknowledgments

Rodent and human data were collected in collaboration with Robert E. Hampson, Brent Roeder, and Sam A. Deadwyler at Wake Forest Baptist Medical Center; Charles Y. Liu, Brian P. Lee, George Nune, and Christianne N. Heck at Keck Hospital of the University of Southern California; and Angelica Nguyen, Susan S. Huang, and Tory H. Gong at Rancho Los Amigos National Rehabilitation Center. We appreciate the assistance of the technical and nursing staff of the Keck Hospital and Rancho Los Amigos for their assistance.

This study was supported by the Defense Advanced Research Projects Agency Restoring Active Memory (RAM) program, contract N66001-14-C-4016 (REH, SAD, TWB, DS). The views, opinions, and findings expressed are our own and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. government.

References

- Abma, B. (2009). *Evaluation of requirements management tools with support for traceability-based change impact analysis*. Master's thesis, University of Twente, Enschede.
- Arabzadeh, E., Panzeri, S., & Diamond, M. E. (2006). Deciphering the spike train of a sensory neuron: Counts and temporal patterns in the rat whisker pathway. *Journal of Neuroscience*, 26, 9216–9226. 10.1523/JNEUROSCI.1491-06.2006, PubMed: 16957078
- Berens, P., Ecker, A. S., James Cotton, R., Ma, W. J., Bethge, M., & Tolias, A. S. (2012). A fast and simple population code for orientation in primate V1. *Journal of Neuroscience*, 32, 10618–10626. 10.1523/JNEUROSCI.1335-12.2012, PubMed: 22855811
- Berger, T. W., Song, D., Chan, R. H. M., Marmarelis, V. Z., Lacoss, J., Wills, J., . . . Granacki, J. J. (2012). A hippocampal cognitive prosthesis: Multi-input, multi-output nonlinear modeling and VLSI implementation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20(2), 198–211. 10.1109/TNSRE.2012.2189133, PubMed: 22438335
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123–140. 10.1007/BF00058655
- Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E., & Frank, L. M. (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, 14(2), 325–346. 10.1162/08997660252741149, PubMed: 11802915
- Carmena, J. M., Lebedev, M. A., Crist, R. E., O'Doherty, J. E., Santucci, D. M., Dimitrov, D. F., . . . Nicolelis, M. A. L. (2003). Learning to control a brain-machine interface for reaching and grasping by primates. *PLOS Biology*, 1(2), e42. 10.1371/journal.pbio.0000042, PubMed: 14624244
- Cheng, S., Li, M., Fan, J., Shang, Z., & Wan, H. (2021). Decoding route selection of pigeon during goal-directed behavior: A joint spike-LFP study. *Behavioural Brain Research*, 409, 113289. 10.1016/j.bbr.2021.113289, PubMed: 33836168
- Collinger, J. L., Wodlinger, B., Downey, J. E., Wang, W., Tyler-Kabara, E. C., Weber, D. J., . . . Schwartz, A. B. (2013). High-performance neuroprosthetic control by an individual with tetraplegia. *Lancet*, 381, P557–P564. 10.1016/S0140-6736(12)61816-9
- de Boor, C. (1972). On calculating with B-splines. *Journal of Approximation Theory*, 6(1), 5–62. 10.1016/0021-9045(72)90080-9
- Deadwyler, S. A., Bunn, T., & Hampson, R. E. (1996). Hippocampal ensemble activity during spatial delayed-nonmatch-to-sample performance in rats. *Journal of Neuroscience*, 16(1), 354–372. 10.1523/JNEUROSCI.16-01-00354.1996, PubMed: 8613802

- Deadwyler, S. A., & Hampson, R. E. (1995). Ensemble activity and behavior: What's the code? *Science*, 270(5240), 1316. 10.1126/science.270.5240.1316, PubMed: 7481817
- Diamantidis, N. A., Karlis, D., & Giakoumakis, E. A. (2000). Unsupervised stratification of cross-validation for accuracy estimation. *Artificial Intelligence*, 116(1–2), 1–16. 10.1016/S0004-3702(99)00094-6
- Dietterich, T. G. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science*, vol. 1857. Berlin: Springer. 10.1007/3-540-45014-9_1
- Donoghue, J. P. (2002). Connecting cortex to machines: Recent advances in brain interfaces. *Nature Neuroscience*. 10.1038/nn947
- Efron, B., & Tibshirani, R. J. (1997). *An introduction to the bootstrap*. Boca Raton, FL: CRC Press.
- Esfahany, K., Siergiej, I., Zhao, Y., & Park, I. M. (2018). Organization of neural population code in mouse visual system. *eNeuro*, 5(4). 10.1523/ENEURO.0414-17.2018
- Georgopoulos, A. P., Schwartz, A. B., & Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science*, 233(4771), 1416–1419. 10.1126/science.3749885, PubMed: 3749885
- Gerstner, W., & Kistler, W. M. (2002). *Spiking neuron models*. Cambridge: Cambridge University Press.
- Ghosh, S., & Maunsell, J. H. R. (2021). Single trial neuronal activity dynamics of attentional intensity in monkey visual area V4. *Nature Communications*, 12, 2003. 10.1038/s41467-021-22281-2
- Glaser, J. I., Benjamin, A. S., Chowdhury, R. H., Perich, M. G., Miller, L. E., & Kording, K. P. (2020). Machine learning for neural decoding. *eNeuro*, 7(4), 1–16. 10.1523/ENEURO.0506-19.2020
- Hampson, R. E., Pons, T. P., Stanford, T. R., & Deadwyler, S. A. (2004). Categorization in the monkey hippocampus: A possible mechanism for encoding information into memory. In *Proceedings of the National Academy of Sciences of the United States of America*, 101(9), 3184–3189. 10.1073/pnas.0400162101
- Hampson, R. E., Song, D., Robinson, B. S., Fetterhoff, D., Dakos, A. S., Roeder, B. M., . . . Deadwyler, S. A. (2018). Developing a hippocampal neural prosthetic to facilitate human memory encoding and recall. *Journal of Neural Engineering*, 15(3). 10.1088/1741-2552/aaed7, PubMed: 29589592
- Hawkins, J., & Blakeslee, S. (2004). *On intelligence*. New York: Macmillan.
- Heelan, C., Lee, J., O'Shea, R., Lynch, L., Brandman, D. M., Truccolo, W., & Numikko, A. V. (2019). Decoding speech from spike-based neural population recordings in secondary auditory cortex of non-human primates. *Communications Biology*, 2(1) 466. 10.1038/s42003-019-0707-9, PubMed: 31840111
- Hlaváčková-Schindler, K., Paluš, M., Vejmelka, M., & Bhattacharya, J. (2007). Causality detection based on information-theoretic approaches in time series analysis. *Physics Reports*, 441(1). 1–46. 10.1016/j.physrep.2006.12.004
- Hochberg, L. R., Serruya, M. D., Friehs, G. M., Mukand, J. A., Saleh, M., Caplan, A. H., . . . Donoghue, J. P. (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442, 164–171. 10.1038/nature04970, PubMed: 16838014
- Humayun, M. S., Weiland, J. D., Fujii, G. Y., Greenberg, R., Williamson, R., Little, J., . . . De Juan, E. (2003). Visual perception in a blind subject with a chronic

- microelectronic retinal prosthesis. *Vision Research*, 43, 2573–2581. 10.1016/S0042-6989(03)00457-7, PubMed: 13129543
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference of Artificial Intelligence* (vol. 14, pp. 1137–1145).
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. 10.1145/3065386
- Lalor, E. C., & Foxe, J. J. (2010). Neural responses to uninterrupted natural speech can be extracted with precise temporal resolution. *European Journal of Neuroscience*, 21(1), 189–193. 10.1111/j.1460-9568.2009.07055.x
- Lanchantin, J., Sekhon, A., & Qi, Y. (2020). Neural message passing for multi-label classification. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11907 LNAI, 138–163. 10.1007/978-3-030-46147-8_9
- Marmarelis, V. Z., & Orme, M. E. (1993). Modeling of neural systems by use of neuronal modes. *IEEE Transactions on Biomedical Engineering*, 40(11), 10.1109/10.245633
- Mauritz, K. H., & Peckham, H. P. (1987). Restoration of grasping functions in quadriplegic patients by functional electrical stimulation (FES). *International Journal of Rehabilitation Research*, 10, 57–60. 10.1097/00004356-198712005-00010, PubMed: 3509753
- McCullagh, P. (1984). Generalized linear models. *European Journal of Operational Research*, 16(3), 285–292. 10.1016/0377-2217(84)90282-0
- Nicolelis, M. A. L. (2003). Brain–machine interfaces to restore motor function and probe neural circuits. *Nature Reviews Neuroscience*, 4, 417–422. 10.1038/nrn1105, PubMed: 12728268
- Oram, M. W., Xiao, D., Dritschel, B., & Payne, K. R. (2002). The temporal resolution of neural codes: Does response latency have a unique role? *Philosophical Transactions of the Royal Society B: Biological Sciences*, 357(1414). 10.1098/rstb.2002.1113
- Paninski, L., Pillow, J. W., & Simoncelli, E. P. (2004). Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural Computation*, 16, 2533–256. 10.1162/0899766042321797, PubMed: 15516273
- Paugam-Moisy, H., & Bohte, S. (2012). Computing with spiking neuron networks. In G. Rozenberg, T. Bäck, & J. Kok (Eds.), *Handbook of Natural Computing*. Berlin: Springer. 10.1007/978-3-540-92910-9_10
- Pouget, A., Dayan, P., & Zemel, R. S. (2003). Inference and computation with population codes. *Annual Review of Neuroscience*, 26, 381–340. 10.1146/annurev.neuro.26.041002.131112, PubMed: 12704222
- Premchand, B., Toe, K. K., Wang, C., Shaikh, S., Libedinsky, C., Ang, K. K., & So, R. Q. (2020). Decoding movement direction from cortical microelectrode recordings using an LSTM-based neural network. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Piscataway, NJ: IEEE. 10.1109/EMBC44109.2020.9175593
- Puchalla, J. L., Schneidman, E., Harris, R. A., & Berry, M. J. (2005). Redundancy in the population code of the retina. *Neuron*, 48, 403–504. 10.1016/j.neuron.2005.10.017, PubMed: 16269359

- Qian, C., Sun, X., Wang, Y., Zheng, X., Wang, Y., & Pan, G. (2020). Binless kernel machine: Modeling spike train transformation for cognitive neural prostheses. *Neural Computation*, 32, 1863–1900. 10.1162/neco_a_01306, PubMed: 32795229
- Quiroga, R. Q., Reddy, L., Koch, C., & Fried, I. (2007). Decoding visual inputs from multiple neurons in the human temporal lobe, 98, 1997–2007. *Journal of Neurophysiology*. 10.1152/jn.00125.2007, PubMed: 17671106
- Raposo, D., Kaufman, M. T., & Churchland, A. K. (2014). A category-free neural population supports evolving demands during decision-making. *Nature Neuroscience*, 17, 1784–1792. 10.1038/nn.3865, PubMed: 25383902
- Rossi, R. G. A. (2019). *Mathematical statistics: An introduction to likelihood based inference*. Hoboken, NJ: Wiley.
- Schwartz, A. B., Kettner, R. E., & Georgopoulos, A. P. (1988). Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement. *Journal of Neuroscience*, 8, 2913–2927. 10.1523/JNEUROSCI.08-08-02913.1988, PubMed: 3411361
- Shanechi, M. M., Hu, R. C., Powers, M., Wornell, G. W., Brown, E. N., & Williams, Z. M. (2012). Neural population partitioning and a concurrent brain-machine interface for sequential motor function. *Nature Neuroscience*, 15, 1715–1722. 10.1038/nn.3250, PubMed: 23143511
- Shanechi, M. M., Williams, Z. M., Wornell, G. W., Hu, R. C., Powers, M., & Brown, E. N. (2013). A real-time brain-machine interface combining motor target and trajectory intent using an optimal feedback control design. *PLOS ONE*, 8(4), e59049. 10.1371/journal.pone.0059049, PubMed: 23593130
- She, X., Robinson, B. S., Berger, T. W., & Song, D. (2020). Accelerating estimation of a multi-input multi-output model of the hippocampus with a parallel computing strategy. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 2479–2482). Piscataway, NJ: IEEE. 10.1109/EMBC44109.2020.9175490
- Smith, P. W., & Schumaker, L. (1982). Spline functions: Basic theory. *Mathematics of Computation*, 38(158), 652. 10.2307/2007301
- Smyth, P., & Wolpert, D. (1998). Stacked density estimation. In S. Solla, T. Leen, & K. R. Müller (Eds.), *Advances in neural information processing systems*, 11. Cambridge, MA: MIT Press.
- Song, D., & Berger, T. W. (2014). Hippocampal memory prosthesis. In D. Jaeger & R. Jung (Eds.), *Encyclopedia of Computational Neuroscience*. Berlin: Springer.
- Song, D., Chan, R. H. M., Marmarelis, V. Z., Hampson, R. E., Deadwyler, S. A., & Berger, T. W. (2007). Nonlinear dynamic modeling of spike train transformations for hippocampal-cortical prostheses. *IEEE Transactions on Biomedical Engineering*, 54(6), 1053–1066. 10.1109/TBME.2007.891948, PubMed: 17554824
- Song, D., Chan, R. H. M., Marmarelis, V. Z., Hampson, R. E., Deadwyler, S. A., & Berger, T. W. (2009). Nonlinear modeling of neural population dynamics for hippocampal prostheses. *Neural Networks*, 22, 1340–1351. 10.1016/j.neunet.2009.05.004, PubMed: 19501484
- Song, D., Hampson, R. E., Robinson, B. S., Marmarelis, V. Z., Deadwyler, S. A., & Berger, T. W. (2016). Decoding memory features from hippocampal spiking activities using sparse classification models. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 1620–1623). Piscataway, NJ: IEEE. 10.1109/EMBC.2016.7591023

- Song, D., Harway, M., Marmarelis, V. Z., Hampson, R. E., Deadwyler, S. A., & Berger, T. W. (2014). Extraction and restoration of hippocampal spatial memories with non-linear dynamical modeling. *Frontiers in Systems Neuroscience*, 8(May), 1–11. 10.3389/fnsys.2014.00097, PubMed: 24478639
- Song, D., Marmarelis, V. Z., & Berger, T. W. (2009). Parametric and non-parametric modeling of short-term synaptic plasticity. Part I: Computational study. *Journal of Computational Neuroscience*, 26(1), 1–19. 10.1007/s10827-008-0097-3, PubMed: 18506609
- Song, D., She, X., Hampson, R. E., Deadwyler, S. A., & Berger, T. W. (2017). Multi-resolution multi-trial sparse classification model for decoding visual memories from hippocampal spikes in human. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (pp. 1046–1049). Piscataway, NJ: IEEE. 10.1109/EMBC.2017.8037006
- Song, D., Wang, H., Tu, C. Y., Marmarelis, V. Z., Hampson, R. E., Deadwyler, S. A., & Berger, T. W. (2013). Identification of sparse neural functional connectivity using penalized likelihood estimation and basis functions. *Journal of Computational Neuroscience*, 35(3), 335–357. 10.1007/s10827-013-0455-7, PubMed: 23674048
- Stevenson, I. H., Rebesco, J. M., Hatsopoulos, N. G., Haga, Z., Miller, L. E., & Kording, K. P. (2009). Bayesian inference of functional connectivity and network structure from spikes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(3). 10.1109/TNSRE.2008.2010471, PubMed: 19273038
- Sussillo, D., Nuyujukian, P., Fan, J. M., Kao, J. C., Stavisky, S. D., Ryu, S., & Shenoy, K. (2012). A recurrent neural network for closed-loop intracortical brain-machine interface decoders. *Journal of Neural Engineering*, 9, 026027. 10.1088/1741-2560/9/2/026027
- Taylor, D. M., Helms Tillery, S. I., & Schwartz, A. B. (2003). Information conveyed through brain-control: Cursor versus robot. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11, 195–199. 10.1109/TNSRE.2003.814451, PubMed: 12899273
- Thorpe, S., Delorme, A., & Van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Networks*, 32, 715–725. 10.1016/S0893-6080(01)00083-1
- Tu, C. Y., Song, D., Breidt, F. J., Berger, T. W., & Wang, H. (2012). Functional model selection for sparse binary time series with multiple inputs. In *Economic Time Series: Modeling and Seasonality*. London: Chapman & Hall.
- Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7, art. 91. 10.1186/1471-2105-7-91, PubMed: 16504092
- Vreeken, J. (2002). Spiking neural networks, an introduction. *Computing*.
- Warland, D. K., Reinagel, P., & Meister, M. (1997). Decoding visual information from a population of retinal ganglion cells. *Journal of Neurophysiology*, 78, 2336–2350. 10.1152/jn.1997.78.5.2336, PubMed: 9356386
- White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*. 10.2307/1912526
- Williams, A. H., & Linderman, S. W. (2021). *Statistical neuroscience in the single trial limit*. 1–25. Retrieved from <http://arxiv.org/abs/2103.05075>
- Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5, 241–259. 10.1016/S0893-6080(05)80023-1

- Zha, Z. J., Hua, X. S., Mei, T., Wang, J., Qi, G. J., & Wang, Z. (2008). Joint multi-label multi-instance learning for image classification. In *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE. 10.1109/CVPR.2008.4587384
- Zhang, Y., Jia, S., Zheng, Y., Yu, Z., Tian, Y., Ma, S., . . . Liu, J. K. (2020). Reconstruction of natural visual scenes from neural spikes with deep neural networks. *Neural Networks*, 125, 19–20. 10.1016/j.neunet.2020.01.033, PubMed: 32070853
- Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., . . . Sun, M. (2018). Graph neural networks: A review of methods and applications. *arXiv*.
- Zhou, Z.-H., & Zhang, M. (2007). Multi-instance multilabel learning with application to scene classification. In J. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems*, 19 (pp. 1609–1616). Cambridge, MA: MIT Press.

Received March 18, 2021; accepted August 19, 2021.