Finding Kinematics-Driven Latent Neural States From Neuronal Population Activity for Motor Decoding

Min-Ki Kim, Jeong-Woo Sohn, and Sung-Phil Kim

***Abstract*— While intracortical brain–machine interfaces (BMIs) demonstrate feasibility to restore mobility to people with paralysis, it is still challenging to maintain high-performance decoding in clinical BMIs. One of the main obstacles for high-performance BMI is the noise-prone nature of traditional decoding methods that connect neural response explicitly with physical quantity, such as velocity. In contrast, the recent development of latent neural state model enables a robust readout of large-scale neuronal population activity contents. However, these latent neural states do not necessarily contain kinematic information useful for decoding. Therefore, this study proposes a new approach to finding kinematics-dependent latent factors by extracting latent factors' kinematics-dependent components using linear regression. We estimated these components from the population activity through nonlinear mapping. The proposed kinematics-dependent latent factors generate neural trajectories that discriminate latent neural states before and after the motion onset. We compared the decoding performance of the proposed analysis model with the results from other popular models. They are factor analysis (FA), Gaussian process factor analysis (GPFA), latent factor analysis via dynamical systems (LFADS), preferential subspace identification (PSID), and neuronal population firing rates. The proposed analysis model results in higher decoding accuracy than do the others ($>$17% improvement on average). Our approach may pave a new way to extract latent neural states specific to kinematic information from motor cortices, potentially improving decoding performance for online intracortical BMIs.**

***Index Terms*— Kinematics-dependent latent factor, motor decoding, intracortical brain–machine interface, neural trajectory, factor analysis.**

I. Introduction

DIRECT connection of brain to devices has been a subject of great interest for decades, driven by demands as diverse as improving people's quality of life with paralysis and addressing neuroscientific issues. For several decades, many studies have attempted to develop clinically valid intracortical brain–machine interfaces (BMIs) that typically translate primary motor cortical (M1) activity into control signals [1]–[5]. However, the estimation of precise kinematic information from the noisy observations of neuronal population activity remains one of the ongoing challenges in BMIs [6].

Recent studies have revealed that although the neurons' population appears to predict kinematic variables reasonably well, only a small fraction of them is associated with generating movements [7]–[10]. A simulation study on neural networks showed that individual M1 neurons do not encode behavioral covariates with equal weights when they generate firing patterns to activate realistic muscles [10]. Thus, decoding the firing activity of single M1 neurons is probably influenced by some non-task-related neurons in the population.

On the basis of this notion, subsequent studies have disclosed that latent neural states of neuronal population activity inferred by dimensionality reduction methods could enable a more accurate and stable estimation of kinematics [11]–[17]. In an early study of this kind, Churchland *et al.* showed that latent neural states of motor cortical activity obtained by dynamic principal component analysis (jPCA) could systematically decrease the trial-to-trial variability of neural dynamics presented compared to neuronal population activity [15]. These dynamic neural states allow us to understand the temporal patterns of neural states in M1 with fewer influences by the noisy activity of single neurons. However, although visualization of the trajectory of neural dynamics is impressive with the help of jPCA, it is unsuitable for extracting kinematic information in a single trial because it does not have an explicit noise model to exclude the independent variance of single neurons [16]. Alternatively, factor analysis (FA) can provide an alternative way to model noise for the single trial analysis. Santhanam *et al.* demonstrated that FA improves decoding performance by modeling correlated neural variability effectively [12]. Recently, a Gaussian process factor analysis (GPFA) in which FA is combined with a Gaussian kernel-based-temporal smoother was used to characterize the temporal patterns of neuronal population activity better than using FA only [13]. GPFA, in particular, significantly improves decoding performance compared with smoothed neuronal population activity [18]. This suggests that anyone who wants better decoding performance should consider the covarying neuronal activity in the population, which is often embed-

Manuscript received December 2, 2020; revised April 12, 2021 and August 29, 2021; accepted September 17, 2021. Date of publication September 22, 2021; date of current version October 5, 2021. This work was supported by the Brain Research Program (NRF-2016M3C7A1904988) and the Brain Convergence Research Programs (NRF-2021M3E5D2A01019542) of the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT. *(Corresponding authors: Sung-Phil Kim; Jeong-Woo Sohn.)*

Min-Ki Kim and Sung-Phil Kim are with the Department of Biomedical Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, Republic of Korea (e-mail: spkim@unist.ac.kr).

Jeong-Woo Sohn is with the Department of Medical Science, Catholic Kwandong University, International St. Mary's Hospital, Gangneung 25601, Republic of Korea (e-mail: jsohn@ish.ac.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSRE.2021.3114367, provided by the authors.

Digital Object Identifier 10.1109/TNSRE.2021.3114367

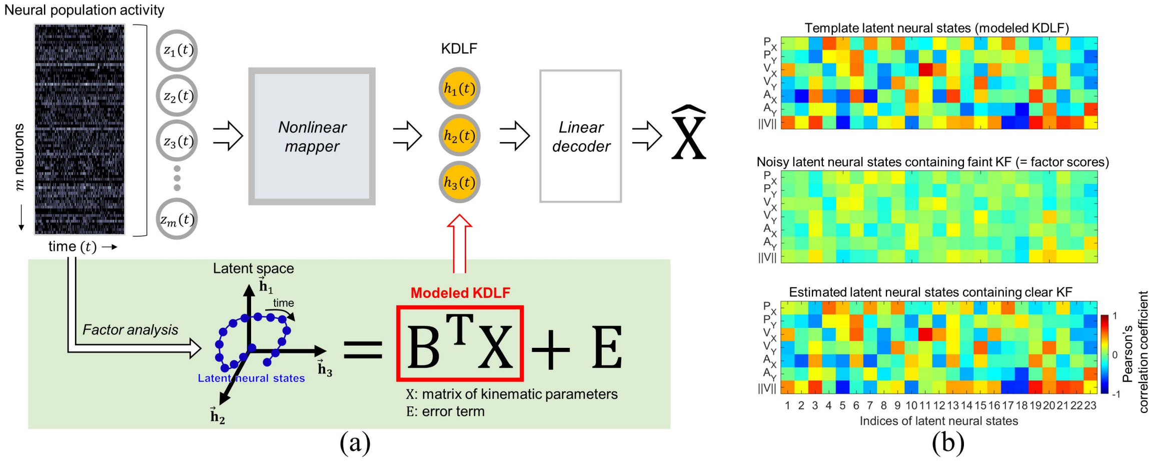

Fig. 1. (a) A schematic illustration of the architecture finding kinematics-dependent latent factor (KDLF) from observed neural population activity. (b) An example of the relationship (correlation coefficients) between kinematic variables and latent neural states (which is from the testing data of monkey C). The top row denotes template latent neural states, corresponding to the modeled KDLF in (a). The middle row denotes noisy latent neural states containing faint kinematic factors (KF), corresponding to factor scores inferred by factor analysis. The bottom row denotes latent neural states containing enhanced KF (estimated KDLF).

ded in latent neural states, rather than only focusing on single neurons for extracting kinematic information more accurately [16].

As the importance of latent neural states has been increasingly highlighted, several researchers have recently begun paying attention to latent neural states' dynamical properties in a single trial [14], [18]–[20]. Latent neural states evolve in time by dynamic rules that govern physical systems during arm movements in most of these studies. Kao *et al.* developed a linear dynamical system model of latent neural states in a single trial [14]. This model captures population activity's dynamical properties and achieves a substantially higher performance of motor decoding than non-dynamical latent neural states. Pandarinath *et al.* recently developed a latent factor analysis via dynamical system (LFADS) to infer de-noised firing rates from noisy spike trains based on complex nonlinear neural dynamics [11]. LFADS modeled neuronal populations' internal dynamics and consequently generated neural states that contain neural state information useful for decoding kinematics. In addition to decoding improvement, these dynamics-based approaches provide neural information that can help validate scientific hypotheses [16].

However, there is no guarantee that these latent neural states would contain specific kinematic information because they find latent neural states via unsupervised learning without kinematic information. In other words, the latent neural state inferred from noisy population neurons is probably blended with less related (or unknown) components to the movement, making it difficult to characterize the latent neural state for kinematics (see the middle row in the Fig. 1(b)). These latent neural states, thus, are likely to end up reflecting poorly on decoding performance.

A recent study demonstrated a preferential subspace identification (PSID) algorithm that can uniquely reveal neural dynamics associated with behavior [21]. PSID is useful for dissociating and modeling behaviorally relevant neural dynamics within the subspace of extracted states. Although these neural dynamics could be helpful ways to describe the complicated dynamical behavior of neural activity, note that these harness the output of dynamical systems and depend on a unique model governing dynamical systems [16].

Therefore, this study proposes an architecture that can further improve latent neural states' kinematic factors with observable behaviors, while preserving intrinsic properties of co-activity as much as possible. To characterize kinematic properties in the latent neural states, which involve inherently overt or covert processes, we built a filter capable of improving kinematic factors from noisy latent neural states by training the defined template for each dimension. Each template (modeled KDLF) is extracted by a linear model coefficient that describes movement information linearly from the inferred neural latent states without considering the noise term (see the top row in the Fig. 1(b)). Considering the benefits of nonlinear approximation capable of describing kinematic parameters precisely from neural activity, we designed one nonlinear mapper that improves latent neural states robustly associated with behaviors through artificial neural networks (ANNs). We believe that not only is this helpful to characterize latent neural states that clarify kinematic properties from neural population activity (see the bottom row in the Fig. 1(b)), but it could also improve BMI performance by applying them to decoding algorithms. Therefore, we propose a new kinematics-driven approach to find neuronal population activity neural states by estimating kinematics-dependent latent factors. Kinematics-dependent latent factors are estimated through two successive steps. First, we perform a typical FA to infer the latent factors of population activity. Second, we extract the components of the latent factors that are accounted for by kinematic variables using linear regression (to define the template for each latent neural factor linearly correlated with kinematic variables). These extracted components are termed kinematics-dependent latent factors (KDLF). Since KDLF can no longer be estimated directly from the population activity, we built a nonlinear mapper to estimate KDLF directly from population activity (i.e., firing rates).

In the proposed approach, we used FA to find initial latent neural states because we can acquire the shared variability of neuronal population activity in a single trial and changes in joint activity of neurons with FA [22]. Since latent neural states can be described as neural signals for generating muscle activations, they are likely to be correlated with various kinematics of limb movements [10], [23]. Thus, we constructed a linear model to relate the latent neural states from FA to several kinematic variables, including position, velocity, acceleration, and speed [15]. We consider this model's output (i.e., KDLF) as latent neural states specific to limb movements' kinematics. To obtain these latent neural states without kinematic information available, as is the case of running BMIs after training, we built a nonlinear mapping between population activity and KDLF using ANNs. First, we compared several nonlinear mappers, which can approximate a mapper for KDLF from neuronal population activity, and then selected one of them to evaluate the effects of KDLF on motor decoding. This study examined the utility of three representative nonlinear mappers to KDLF estimation, namely, multilayer perceptron (MLP) [24], long short-term memory (LSTM) recurrent neural networks (RNNs) [25], and deep canonical correlation analysis (DCCA or deep canonically correlation autoencoder) [26], [27]. MLP, one of the ANNs, consists of multiple hidden layers, including input and output layers [24]. MLP is often used to approximate nonlinear models between neuronal population activity and kinematic variables, as it has a simple structure compared to the other nonlinear mappers. However, the vanishing gradient problem, one of the chronic problems of MLP, makes it challenging to estimate sophisticated nonlinear models. LSTM, another version of RNNs, allows high-performance sequential data processing without the vanishing gradient problem [28]. It could also help predict kinematic information from M1 activity, but it requires many data and a long training time. Meanwhile, DCCA, which finds optimal parameters to maximize the correlation between latent variables for two random variables is often used for different purposes other than MLP or LSTM [27]. Kim *et al.* showed the effects of latent variables extracted by DCCA on motor decoding performance, where the two random variables correspond to firing rates and kinematic variables [29]. This examined the feasibility of motor decoding with DCCA structure-extracting latent variables by directly associating neuronal population activity with kinematic variables. In contrast, our proposed approach focused on finding KDLF that enhances kinematic characteristics to the latent neural states inferred by the FA. Namely, DCCA for KDLF referred to in this study was the only example of the nonlinear mapper that improves the signal-to-noise ratio of the latent neural states, so it should rather focus on the structure for estimating KDLF. Note that each trained nonlinear mapper was only used to estimate KDLF from neural activity and compared with each other, and not all the mappers were used in the decoding phase. In other words, we focused on the effects of structural procedure for estimating KDLF on motor decoding rather than the ability of these nonlinear mappers in this study. Through this structural procedure, we hypothesize that estimated KDLF by the nonlinear mapping would contain kinematic information enough for high-performance motor decoding.

We used datasets of three non-human primates to perform independent and dissimilar arm reaching movement tasks on two- or three-dimensional spaces to validate this hypothesis. We examine how well KDLF represents kinematics by visualizing its trajectories via jPCA [15]. We also predicted various kinematic parameters of arm movements from KDLF by a simple linear decoder and compared the decoding performance with three other well-known latent neural states obtained by FA and GPFA, LFADS, and PSID.

II. Materials and Methods

A. Experimental Animals and Behavioral Tasks

Three independently obtained datasets of rhesus monkeys (*Macaca mulatta*) were used for this study (monkeys C, M, and F). The monkeys were implanted with a 96-channel intracortical microelectrode array (MEA) (Utah Array, Blackrock Microsystems, Salk Lake City, USA). The MEAs were commonly implanted in the primary motor cortex (M1) arm area in all three monkeys. An additional MEA was implemented in the dorsal premotor cortex (PMd) in monkey M or in the ventral premotor cortex (PMv) in monkey F. Yet only neuronal activity data recorded from the MEA in M1 were used in this study. All the surgery and animal care procedures were approved by the Institutional Animal Care and Use Committee at Northwestern University (monkeys C and M) and the University of Pittsburgh (monkey F). Note that the datasets of monkey C [30] and M [31], [32] are publicly available in the repository of Database for Reaching Experiments and Models and a Collaborative Research in Computational Neuroscience.

Monkey C was trained to perform a 2D center-out reaching task by controlling a two-link manipulandum. Monkey F was trained to perform a 3D center-out reaching task in a virtual reality environment. The detailed experimental procedures were described elsewhere for monkey C [30] and monkey F [33], [34], respectively. Briefly, monkeys C and F were instructed to move the computer cursor from a starting location toward one of the radial targets equidistant from the center. For monkey C, eight targets spaced at 45° intervals over a circle were presented. For monkey F, 26 targets were presented, evenly distributed on the surface of a spherical working space. In monkey F, the arm position was monitored and recorded with optical tracking systems (Northern Digital Inc., Waterloo, Canada).

Monkey M was trained to perform a randomly sequential reaching task in 2D spaces by controlling a two-link manipulandum. The detailed experimental procedures are described in [31]. Briefly, monkey M was instructed to move the computer cursor from a starting location toward a pseudo-randomly located target. In contrast to monkey C and F's task reaching one target in a trial, this task presented four targets sequentially in a trial and, thus, could be more naturalistic [28].

In this study's datasets, monkeys C, M, and F performed 175, 498, and 468 trials, respectively. Furthermore, bad trials in which tasks were failed or kinematic data were contaminated by noise were excluded, leaving 175, 358, and 451 trials in monkeys C, M, and F, respectively, for our data analysis.

B. Data Recording

Single neuronal units were recorded using different neural acquisition systems for each monkey: Cerebus (Blackrock Microsystems, Salk Lake City, USA) in monkeys C and M and OmniPlex (Plexon Inc, Dallas, Texas, USA) in monkey F. In monkeys C, M, and F, 172, 67, and 143 single units were sorted, respectively. Furthermore, we excluded silent units with sparse (units corresponding to the mean firing rate in a single trial <5% in the distribution of all trials for the mean firing rates within a single trial) or no spikes, only analyzing 101, 48, and 79 units in monkeys C, M, and F, respectively. Note that we observed no significant difference between decoding performance before and after discarding the single units with quiescent (except single units with no spikes) through preliminary decoding test with the linear decoder ($p < 0.05$, Wilcoxon rank-sum test).

Each unit's neural spike train over the entire recordings in every monkey was binned by a 20-ms time bin without overlapping to estimate instantaneous firing rates. We then conducted smoothing by the square root of spike counts with a Gaussian kernel, where the kernel standard deviation was set to 30 ms. The hand position of each monkey was also resampled to match the resolution of the spike counts (i.e., 20 ms) using the piecewise cubic Hermite interpolating polynomial, which was implemented in MATLAB (Mathworks Inc., Natick, MA, USA) function, "pchip." Then, we extracted an epoch of the neuronal and hand position data corresponding to each reaching trial, which was defined as from 160 ms before a target cue to the time of target acquisition.

C. Extracting Kinematics-Dependent Latent Factors

We initially obtained d-dimensional latent factors, $\mathbf{h} = \{h_1, \ldots, h_d\}^{\mathrm{T}} \in \mathbb{R}^{d\times 1}$, of the m-dimensional smoothed spike counts (SSC), $\mathbf{z} = \{z_1, \ldots, z_m\}^{\mathrm{T}} \in \mathbb{R}^{m\times 1}$, by FA based on the following generative model:

$$\mathbf{z} = \mathbf{F}\mathbf{h} + \mathbf{c} + \mathbf{e} \tag{1}$$

where $\mathbf{F}$ is a matrix of factor loadings, $\mathbf{e}$ denotes an error term following a Gaussian distribution, and $\mathbf{c}$ is the mean SSC vector calculated from all data samples. The number of factors for two approaches (FA and GPFA) to be compared in this study was equally fixed, except for LFADS and PSID, which were optimized by sweeping the hyperparameters (described in section II. E), and was empirically configured as follows: 19 for monkey C, 15 for monkey M, and 28 for monkey F. To evaluate whether the chosen number of dimensions is reasonable, we traced changes in log-likelihoods inferred from validation-testing data with the FA model trained by increments by one from 2 to 35, where the maximum log-likelihood reflects co-activity of population neurons [22] (see Fig. S1 in the Supplementary Materials). Fig. S1 in the Supplementary Materials depicts changes in log-likelihoods over d in the FA model in each dataset. As a result, we confirmed that the number of dimensions was around the maximum log-likelihood. These procedures were conducted through fourfold cross-validation from the training set.

Next, we modeled the resulting latent factor, $\mathbf{h}$, by a kinematic vector $\mathbf{x}$ that included several types of kinematic parameters: position $\mathbf{p} \in \mathbb{R}^{2\ \text{or}3}$, velocity $\mathbf{v} \in \mathbb{R}^{2\ \text{or}3}$, acceleration $\mathbf{a} \in \mathbb{R}^{2\ \text{or}3}$, and speed $\|\mathbf{v}\| \in \mathbb{R}^1$. A linear regression model was used to explain the ith variable in $\mathbf{h}$by $\mathbf{x}$ as follows:

$$h_i\,(t) = \boldsymbol{\beta}_i \mathbf{x}\,(t) + \epsilon_i(t), t = 1, \ldots, T \tag{2}$$

where $h_i(t)$ denotes the ith variable in $\mathbf{h}$ at time instant t, $\boldsymbol{\beta}_i$ indicates a vector of linear model coefficients for the ith variable, and $\epsilon_i(t)$ is the error term following Gaussian distribution. We estimated $h_i\ (t)$ for every i using individual linear regression models. Here we used as many kinematic parameters as possible because latent neural states are tightly correlated with muscle activation patterns [15], [23]. Finally, we defined the model outputs, $\hat{h}(i)$ for $i = 1, \ldots, d$, as KDLF. We regarded KDLF as the components of original latent factors that are accounted for by all kinematic parameters.

D. Estimating Kinematics-Dependent Latent Factors From Population Activity

To estimate KDLF from neuronal population activity, we designed a nonlinear mapper between $\mathbf{z}$ and $\mathbf{h}$(see Fig. 1(a)). We approximated this mapper with three ANNs, namely, MLP, DCCA, and LSTM, and compared Pearson's correlation coefficient (CRR) distributions between the modeled KDLF and the estimated KDLF. The nonlinear function approximations were conducted to minimize the cost function between input neuronal population activity ($\mathbf{z}$) and those of KDLF outputs ($\mathbf{h}$), such as

$$\{\boldsymbol{\Theta}^{*}, \mathbf{b}^{*} = \arg\min_{\Theta,\mathbf{b}} f_e(\Theta^T\mathbf{z} + \mathbf{b}, \mathbf{h}) \tag{3}$$

where Θ is a matrix of the weights, b is a vector of bias of the networks, and $f_e(\cdot)$ denotes the cost function. Assuming nonlinear relationships between $\mathbf{z}$ and $\mathbf{h}$, we can estimate KDLF $\hat{h}_t$ from $\mathbf{z}_t$ at time instant t:

$$\hat{\mathbf{h}}_t = f\left(\mathbf{z}_t; \Theta^{*}, \mathbf{b}^{*}\right), \tag{4}$$

where $f(\cdot)$ indicates the trained nonlinear mapper by ANNs.

All ANNs were fixed to two layers, where the hyperparameters were optimized through the Bayesian optimization algorithm ("bayesopt" package in MATLAB), which was repeated by 300 [35]. MLP, which is one of the ANNs and consists of multiple layers with an input, output, and multiple hidden layers, was configured with a "feedforwardnet" package in MATLAB and optimized four parameters within specific ranges with fixed hyperbolic tangent transfer function. The range of each hyperparameter is as follows: the number of hidden nodes $\{5, 6, \ldots, 100\}$, the minimum gradient $\{1e-8, \ldots, 1e-5\}$, learning rates $\{0.001, \ldots, 0.1\}$, learning rate of decrease $\{0.01, \ldots, 0.5\}$, and increase ratio $\{1, 2, \ldots, 15\}$. DCCA, which finds parameters to maximize the correlation between canonical variables for two random variables ($\mathbf{z}$ and $\mathbf{h}$) based on the autoencoder structure, was configured using a MATLAB package available at Weiran Wang's website (https://ttic.uchicago.edu/~wwang5/dccae.html) [27]. For DCCA, we optimized ten hyperparameters with fixed linear rectifier function within ranges as follows: the number

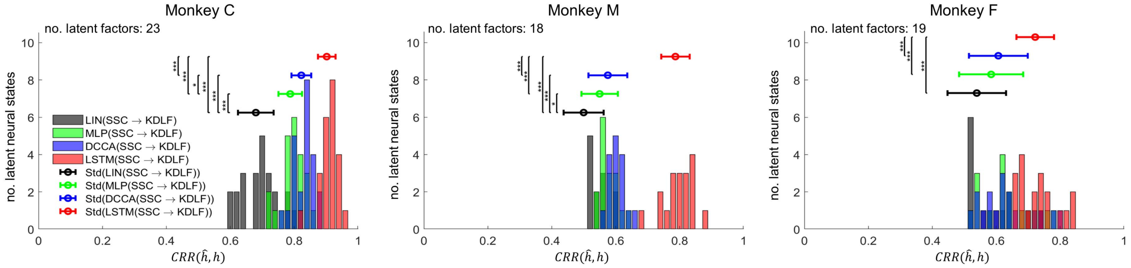

Fig. 2. Distribution of correlation coefficients (*CRR*) between modeled and estimated KDLF for each monkey. Each color denotes types of linear or nonlinear mappers. Horizontal error bars denote standard deviation of *CRR* across the total latent neural states for each mapper. Asterisks with a vertical solid line denote statistically significant difference, which is based on the Kruskal-Wallis test and a post hoc analysis with Bonferroni correction, $^{***}p < 0.001$, $^{*}p < 0.05$.

of hidden nodes $\{5, 6, \ldots, 512\}$ for $\mathbf{z}$ and $\mathbf{h}$, regularization parameters $\{1e-6, \ldots, 1e-3\}$ for $\mathbf{z}$ and $\mathbf{h}$, the mini-batch size for estimating gradient $\{16, 17, \ldots, 1024\}$ for $\mathbf{z}$ and $\mathbf{h}$, weight decay parameters for all weight parameters $\{1e-6, \ldots, 1e-3\}$, the initial learning rate $\{0.001, \ldots, 0.1\}$, the learning rate decays $\{0.5, \ldots, 1\}$, and the momentum parameter $\{0.5, \ldots, 1\}$. LSTM, which is included in deep learning fields, is known to be effective in processing single and sequential data [25], [28]. For LSTM, we optimized eight hyperparameters within ranges as follows: the number of hidden nodes $\{5, 6, \ldots, 100\}$, the initial learning rate $\{0.001, \ldots, 0.1\}$, the minibatch size $\{16, \ldots, 1024\}$, L^2-regularization parameters $\{1e-6, \ldots, 1e-3\}$, the gradient decay factor $\{0.5, \ldots, 1\}$, squared gradient decay factor $\{0.5, \ldots, 1\}$, learning rate drop factor $\{0.5, \ldots, 1\}$, and learning rate drop period $\{5, 6, \ldots, 100\}$. The optimized hyperparameters of MLP, DCCA, and LSTM for all monkeys were listed in Table S1 in the Supplementary Materials. We also measured the training time for each nonlinear mapper with the optimized hyperparameters as summarized in Table S3 of the Supplementary Materials.

Optimized nonlinear mappers will estimate KDLF from neuronal population activity according to structural characteristics for each ANN. To choose one appropriate method, we computed the element-wise Pearson's CRRs between $\mathbf{h}$ and $\hat{\mathbf{h}}$. Fig. 2 depicts the CRR distributions for each monkey. Thus, we examined the effects of KDLF on motor decoding by selecting the LSTM with the highest average CRR across all elements (dimensions).

E. Other Latent Neural States

We conducted a comparative analysis between neural states formed by KDLF and four other existing neural states obtained by SSC, FA, GPFA, and LFADS. Note that neural states by smoothing spike counts as described above kept the original population activity without further inferring latent variables (i.e., $d = m$). For neural states by FA, we used latent factors inferred in the above procedure, that is, $\mathbf{h}$ in the previous section.

GPFA is a variant of FA to minimize the noise characteristics with a Gaussian kernel smoothing process in a single trial [13]. Similarly, as in FA, we empirically found optimal dimensionalities for GPFA with the minimum reconstruction error for all monkeys: 10 for monkey C, 19 for monkey M, and 15 for monkey F. We initialized the Gaussian process (GP) timescale to 100 ms and GP noise variance to 0.001. Neural states estimated by FA and GPFA were used as input to a motor decoding algorithm described below.

LFADS infers de-noised firing rates by processing the underlying dynamics of neuronal population-spiking activity [11]. It models a generic dynamical system through a variational autoencoder based on RNNs to describe population activity dynamics. We used LFADS Run Manager available in GitHub (https://lfads.github.io/lfads-run-manager, written by Daniel J. O'Shea) and inferred de-noized firing rates on the basis of the guidelines provided in [11]. In this study, we conducted only single LFADS, not the stitched LFADS. Hyperparameters for each monkey were mainly referenced in the second row of Supplementary Table I in ref [11]. However, we swept several parameters, including the number of units in the encoder and generator, because of different input units, even though the number of trials is similar to the referenced parameter data. The number of units in the encoder is 128, 64, and 64 for monkeys C, M, and F, respectively, and the number of units in the generator is 32, 32, and 256. Then, we set optimal number of latent factors by measuring decoding performance, while sweeping latent factors. The number of latent factors is 28, 24, and 28 for monkeys C, M, and F, respectively. Meanwhile, bin sizes corresponding to such important hyperparameter were similarly configured to FA.

The PSID algorithm is useful to dissociate and model behaviorally relevant neural dynamics. It models beneficial latent neural states with time-series datasets (neural activity and behavioral data) and state dimension parameters by identifying a general dynamic linear state space model's parameters, while considering behaviorally relevant neural dynamics [21]. We used a PSID package in the MATLAB algorithm that is available online at https://github.com/ShanechiLab/PSID and optimized the parameters for PSID under the constraints specified in the toolbox description. Therefore, we set the state dimensions as 35, 34, and 24, for monkeys C, M, and F, respectively, by measuring decoding performance, while sweeping state dimensions by one from 2 to 35 (see Fig. S3 in the Supplementary Materials). Notably, all hyperparameter optimization was conducted through fourfold cross-validation, with the validation data separated from the training data.

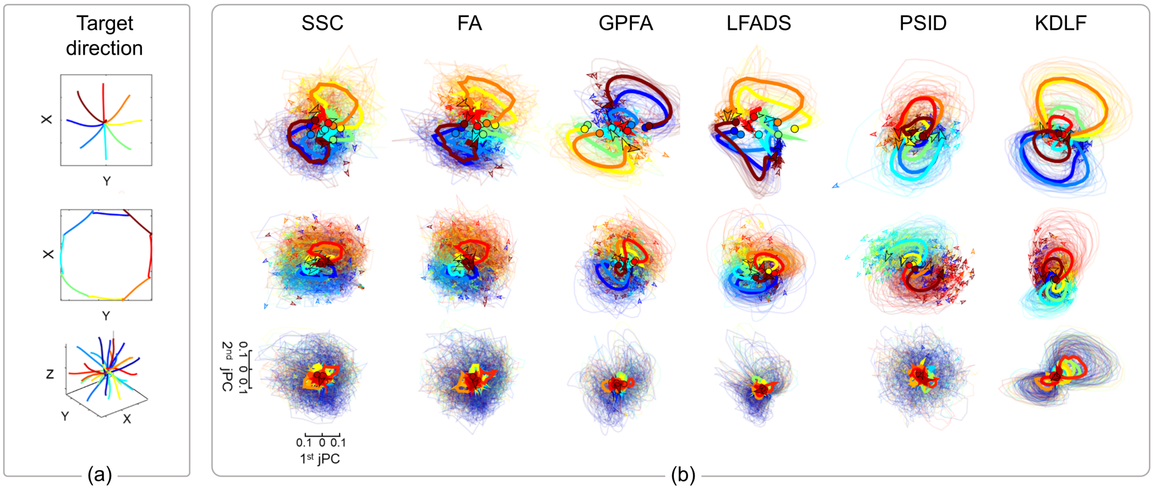

Fig. 3. Visualization of jPCs' trajectories for latent neural states obtained by each different approach (testing data). The panel (a) indicates averaged cursor movement trajectories for each monkey, and each color denotes a target. Note that the target directions in monkey M were binned as eight-angle widths uniformly distributed in an interval of 45° for each sub-trial. The panel (b) denotes jPCs' trajectories. The top to bottom rows corresponds to monkey C, M, and F, respectively. A thin colored line corresponds to the jPC trajectory in a single trial, and a thick colored line is the average jPC trajectories across trials for each target. Solid circle denotes motion onset.

F. Neural Trajectory Visualization

We first compared the trajectories of the neural dynamics of all five neural states (KDLF, SSC, FA, GPFA, and LFADS) using jPCA. jPCA visualizes the output-null model representing the temporal trajectory of latent factors of neuronal population activity [15], [23]. Especially, the visualized trajectories reflect the dynamical properties of neuronal population activity and can capture the influence of behavioral status better than a single neuronal firing activity. To generate neural trajectories, we utilized the jPCA toolbox available for download from the Churchland lab homepage (https://churchland.zuckermaninstitute.columbia.edu).

G. Decoding and Evaluation

We decoded neural states generated by each method above to predict four kinematic parameters (position, velocity, acceleration, and speed) using a simple linear decoder, given by

$$\mathbf{x}(t) = A\mathbf{y}(t) \tag{5}$$

where A denotes a matrix of the decoder coefficients, $\mathbf{x}(t)$ is the kinematic parameter vector, $\mathbf{y}(t)$ is the given neural state vector because of the analysis models for comparison. For each dataset, the first 75% of all trials were set as a training set, and the remaining 25% were used for testing (training set = 132 trials, testing set = 43 in monkey C; training set = 269, testing set = 89 in monkey M; and training set = 339, testing set = 112 in monkey F). We did not perform cross-validation decoding in this study because we emulated an online BMI situation where training and testing are conducted in sequence, although we conducted fourfold cross-validation in a separate analysis and confirmed that the decoding evaluation results were not significantly different between the sequential testing and cross- validation. Notably, biased results may occur depending on each dataset's characteristics. To evaluate the decoding performance, we measured root-mean-square error (RMSE) and CRRs between the predicted and actual values of each kinematic parameter.

III. Experimental Results

A. Estimation of Kinematics-Dependent Latent Factors

We first verified how well mappers predicted KDLF from neuronal population activity. Fig. 2 depicts the distributions of element (dimension)-wise CRRs between modeled KDLF and KDLF estimated using linear or nonlinear mappers. Here, we further compared nonlinear mappers with linear mappers (linear regression).

We observed that KDLF estimated by LSTM revealed significantly higher CRRs than other mappers (Kruskal-Wallis test and a post hoc analysis with Bonferroni correction, $p < 0.05$). The mean CRR of LSTM was 0.91 ± 0.03 in monkey C, 0.78 ± 0.04 in monkey M, and 0.74 ± 0.06 in monkey F. Also, DCCA (0.82 ± 0.03 in monkey C, 0.58 ± 0.06 in monkey M, and 0.61 ± 0.09 in monkey F) and MLP (0.77 ± 0.06 in monkey C, 0.55 ± 0.06 in monkey M, and 0.59 ± 0.09 in monkey F) followed by a better performance than linear models (0.68 ± 0.05 in monkey C, 0.50 ± 0.06 in monkey M, and 0.54 ± 0.09 in monkey F), and they showed a statistically significant difference. These results demonstrate that we could reliably estimate KDLF from the population activity using LSTM, a nonlinear mapper.

B. Dynamic Rotation Properties of Latent Neural States

We visually observed each method's neural dynamics (SSC, FA, GPFA, LFADS, PSID, and KDLF). To this end, we projected all neural states onto a 2D space using jPCA. Then, we visualized each method's neural trajectories in the 2D spaces where each trajectory was specified by target directions (Fig. 3). We also marked motion onset on every trajectory (solid circles in Fig. 3). Note that each motion onset was defined as a time point corresponding to 15% of the maximum

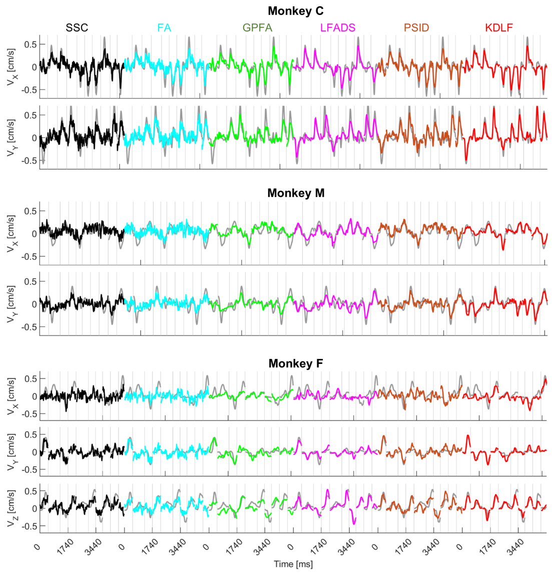

Fig. 4. Examples of actual and predicted velocity trajectories for testing data (ten trials). Gray line denotes actual velocity in example trials of the testing set, and each colored line indicates the trajectory of the predicted velocity of segmented by a vertical dashed line according to five neural states: SSC (dark), FA (cyan), GPFA (green), LFADS (magenta), PSID (orange), and KDLF (red). Each vertical line represents the boundary between trials.

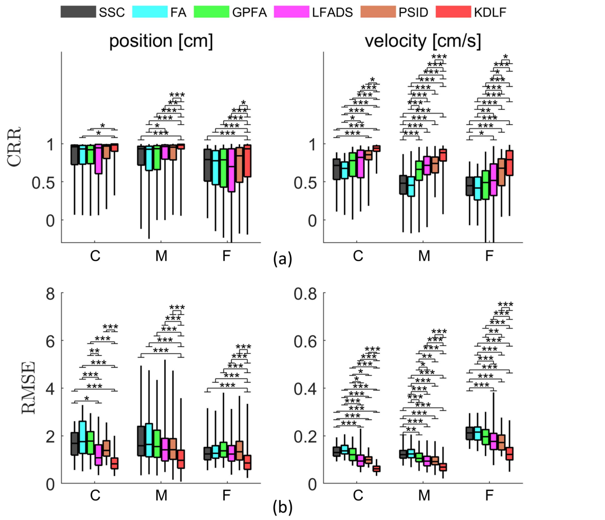

Fig. 5. Decoding performances, which are based on testing data in each monkey (C, M, and F). (a) indicates CRR between actual and predicted outputs for position, and velocity, and (b) denotes RMSE. Horizontal lines on each boxplot correspond to 25 %, 50 %, and 75 % of the distribution of CRRs or RMSEs. Each vertical line indicates the minimum and maximum CRR or RMSE. Asterisks with a horizontal line denote significant difference between latent neural states (Kruskal-Wallis test and a post hoc analysis with Bonferroni correction, $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$).

speed [36], where we checked the appropriacy for the setting of motion onset with the actual hand position data in every trial of each subject and confirmed that it could sufficiently describe each reaching movement. The mean motion onset was 269 ± 64 ms after target cues in monkey C, 106 ± 73 ms in monkey M, and 162 ± 63 ms in monkey F.

During movements, all latent neural states' neural trajectories swept through both dimensions of jPC1 and jPC2, consistent with patterns reported in [23] (Fig. 3). In particular, the neural trajectories of KDLF appeared to show more consistent and clearer rotational patterns than do other latent neural states. Also, the neural trajectories of KDLF tended to be more spatially organized such that adjacent neural trajectories corresponded to adjacent target directions.

C. Motor Decoding

We evaluated the performance of decoding six latent neural states into various kinematic parameters with a simple linear decoder built and tested in each monkey. Here, we predicted four kinematic parameters (position, velocity, acceleration, and speed). As an example, Fig. 4 illustrates the trajectories of actual velocity along with those of predicted velocity for all latent neural states (SSC, FA, GPFA, LFADS, PSID, and KDLF). In this example, predicted velocities decoded from KDLF were closer to actual ones than were those from other latent neural states over different monkeys. We then measured ACCs for velocity and position (Fig. 5(a-b)) across working space axes and the RMSEs for the velocity and position (Fig. 5(c-d)) representing the decoding performance for six neural representations, including SSC. We conducted a statistical process with a Kruskal-Wallis test, where independent variables consisted of latent neural states (i.e., SSC, FA, GPFA, LFADS, PSID, and KDLF). For velocity decoding, the Kruskal-Wallis test showed the main effect of latent neural states on CRR ($\chi^2 = 84.5$, $p < 0.01$ in monkey C; $\chi^2 = 298.9$, $p < 0.01$ in monkey M; and $\chi^2 = 159.1$, $p < 0.01$ in monkey F) as well as RMSE ($\chi^2 = 140.3$, $p < 0.01$ in monkey C; $\chi^2 = 267.4$, $p < 0.01$ in monkey M; $\chi^2 = 228.9$, $p < 0.01$ in monkey F). The post hoc multiple-comparison test with Tukey–Kramer correction revealed that KDLF yielded better performance than did PSID, LFADS, GPFA, FA, and SSC for all monkeys on CRR (see Fig. 5(a)) and RMSE (see Fig. 5(b)). KDLF yielded a 13.3% increase in decoding performance on average compared to the second best latent neural states by PSID (monkey C: 15.3%, monkey M: 12.0%, and monkey F: 12.6%), and LFADS (monkey C: 17.9%, monkey M: 14.4%, and monkey F: 19.3%).

We next reconstructed the position trajectory by integrating the predicted velocity and compared it among latent neural states (see Fig. 6). It demonstrated that the reconstructed trajectories from KDLF traced actual ones more accurately than did those from other latent neural states. This was especially well pronounced in the 3D movements in monkey F. For reconstructed position parameter, the Kruskal-Wallis test showed the main effect of latent neural states on CRR ($\chi^2 = 14.9$, $p < 0.05$ in monkey C; $\chi^2 = 63.5$, $p < 0.01$ in monkey M; and $\chi^2 = 41.4$, $p < 0.01$ in monkey F) as well as RMSE ($\chi^2 = 71.0$, $p < 0.01$ in monkey C; $\chi^2 = 73.2$, $p < 0.01$ in monkey M; $\chi^2 = 86.9$, $p < 0.01$ in monkey F). The post hoc multiple-comparison test with Tukey–Kramer correction revealed that KDLF yielded better performance than did PSID, LFADS, GPFA, FA, and SSC in most monkeys, excluding comparison with LFADS in monkey C on CRR (see Fig. 5(a)) and RMSE (see Fig. 5(b)). KDLF yielded a 11.9% increase in decoding performance on average (across CRR and RMSE) compared to the second best latent neural

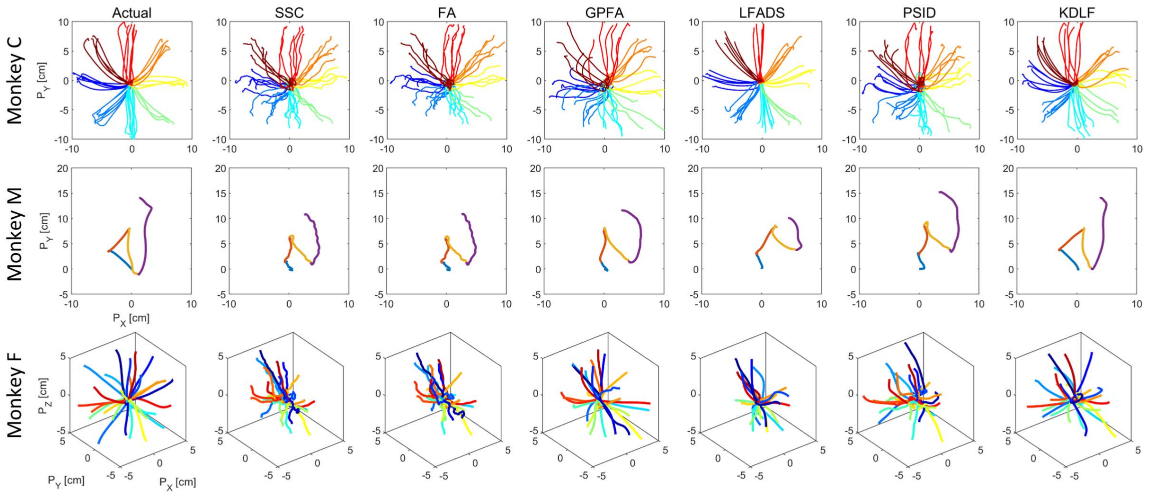

Fig. 6. Examples of position trajectories reconstructed from predicted velocity for testing data. Each colored line denotes one trial for a specific target direction. For monkey C, colors denote eight radial targets, and trajectories correspond to 43 trials. For monkey M, colors correspond to four sequential sub-trials within a single trial. For monkey F, colors indicate 26 radial targets, and one trial per target is shown.

states by PSID (monkey C: 14.4%, monkey M: 9.9%, and monkey F: 11.4%), and LFADS (monkey C: 11.0%, monkey M: 11.5%, and monkey F: 17.8%). As with position and velocity decoding, KDLF yielded significantly higher performance than PSID and LFADS for acceleration and speed either (see Fig. S12 in the Supplementary Materials). Additional statistical statements for acceleration and speed parameters are described in Fig. S12 and Table S2 of the Supplementary Materials.

IV. Discussion

This study proposed a new approach for finding kinematics-dependent latent neural states of M1 population activity. We estimated kinematics-dependent latent neural states by nonlinearly mapping M1 population firing rates to KDLF, where the nonlinear mapper was approximated by LSTM. The reason we explore latent factors first is because latent factors inferred by factor analysis have been demonstrated to represent neural population activity in M1 reasonably well by neuroscience literature (e.g., [22], [37], [38]). We thus opt to use this well-known approach to find latent variables. It also provides us an opportunity to directly compare with similar state-of-the-art methods such as LFADS, GPFA and PSID that are based on the same latent factors or a similar linear subspace method. We thus validated the proposed latent neural states by examining the neural dynamics information presented in their neural trajectory and by decoding the latent neural states into various kinematic parameters. We compared KDLF with other latent neural states obtained by SCC, FA, GPFA, PSID, and LFADS. We found that the neural trajectories produced by KDLF revealed neural dynamics after movement onset more clearly than those by other methods (see Fig. 3 and Figs. S2 and S4 in the Supplementary Materials) and that decoding KDLF enabled a simple linear decoder to predict kinematic parameters more accurately than decoding other neural presentations (Figs. 4–6). Notably, KDLF showed consistently high decoding performance for different monkeys performing dissimilar tasks.

To validate the decoding performance of the proposed approach, we predicted four kinematic parameters from KDLF, including position, velocity, acceleration, and speed. We compared the decoding performance of KDLF with those of five other latent neural states, including SSC (i.e., SSC, FA, GPFA, LFADS, and PSID). Decoding KDLF yielded higher performance for predicting velocity and acceleration than did all other counterparts and slightly higher performance for predicting speed than did LFADS, the second-best model. Notably, this study did not attempt to investigate the effect of using multiple stitched sessions, in which LFADS has shown advantages for maintaining decoding performance. Furthermore, it may be difficult to argue that it is substantially better than LFADS since it does not optimize the bin width (fixed to 20-ms bins) among the important hyperparameters of LFADS. However, it could be a considerable advantage to improve the behaviorally relevant latent neural states with a few parameters.

Contrary to expectations, PSID did not show a dramatic performance improvement compared to LFADS for all monkeys (see Fig. 5(c-d)), even though the specified latent state dimensions reached the best achievable performance by optimizing the number of latent states. The difference between [21] and this study is that this study utilized the datasets composed of kinematic information from the end point rather than angular velocities for an arm. Therefore, the decoding result in this study suggests that the proposed latent neural states of KDLF would improve motor decoding for intracortical BMIs.

According to [15], neural trajectories of a complex kinematic model present clear rotational patterns for behavioral events (e.g., reaching a specific target in a single trial). We observed that the neural trajectories produced by KDLF dynamically vary by sweeping the jPC space during arm movement after motion onset. These characteristics of neural trajectories associated with arm movements were most pronounced with KDLF than they were with other latent neural states. Thus, we speculate that a more reliable and clear representation of movements in KDLF might be linked to improved decoding.

Latent neural states are related to generating muscle activation patterns involved in complex movements [10], [16], [23]. For instance, muscles that generate a specific movement would be driven by a linear combination of neural activities [23]. However, since it is difficult to predict complex kinematic information from noisy neural manifolds with a simple linear model, a nonlinear model, such as deep neural networks (DNNs), has often been proposed [11], [39], [40]. Here, we estimated KDLF by selecting LSTM, one of the three types of DNNs (MLP, LSTM, and DCCA), to approximate nonlinear mapping between neuronal population activity and KDLF that is supposed to represent various kinematic information. We observed that LSTM could yield significantly higher performance than could MLP and DCCA in terms of the CRRs between modeled and estimated KDLF (13.2% higher on average).

In this regard, a framework that directly finds the vector of hidden layer nodes in LSTM could be considered while direct mapping from neural population activities to kinematic variables. We first compared KDLF with the vector of hidden layer nodes in LSTM, which predicts kinematic variables from neural population activities (called "direct LSTM" hereafter). We observed that there was no significant difference between them in terms of decoding performance for each of three monkeys (Kruskal-Wallis test and the post hoc analysis with Bonferroni correction, Fig. S9 in the Supplementary Materials). In addition, we compared KDLF with direct LSTM, and as with the framework above, there was no significant difference for all monkeys (Kruskal-Wallis test and the post hoc analysis with Bonferroni correction, see Fig. S11). Further, KDLF showed a slightly better performance than LSTM in terms of RMSE: for position decoding in the monkeys C and M (for position, $p_{\text{monkeyC}} = 0.02$ (by 9.8%), and $p_{\text{monkeyM}} = 0.02$ (by 8.0%); for velocity, $p_{\text{monkeyM}} = 0.03$ (by 4.6%); and for speed, $p_{\text{monkeyM}} = 0.001$ (by 8.2%) (Kruskal-Wallis test and the post hoc analysis with Bonferroni correction, Fig. S11 in the Supplementary Materials). However, there was no significant difference in terms of CRR.

Although finding the node vector of hidden layers in the direct LSTM allows a straightforward architecture and yielded comparable decoding performance to KDLF, when we evaluated the latent neural states created by this new machine-learning based approach, the neural trajectories by jPCS from these latent neural states could not represent kinematics well compared to those from KDLF (see Fig. S5(b) and S6 in the Supplementary Materials). Specifically, we observed that jPCs of KDLF captured higher variance than those of the node vector of hidden layer in the direct LSTM did, where such variance is known to reflect the strength of dynamic patterns that are present in the data (see Fig. S6 in the Supplementary Materials). It shows that using a nonlinear approximator approach to find latent states yielded less meaningful latent variables than our proposed approach. In other words, because the key difference between them is whether based on latent factors or not apart from LSTM use, in this context, KDLF may be more intuitive and advantageous than the node vector of hidden layer in direct LSTM for tracking input neurons associated with a specific kinematic component such as the movement speed (e.g., Fig. S13 in the Supplementary materials).

In addition, we computed variance accounted for (VAF) as a measure of the variance of original neural population activity explained by each latent neural state (FA, GPFA, LFADS, PSID, and KDLF). We used the coefficient of determination (r^2) of multiple linear regression to estimate VAF. As expected, KDLF showed a lower VAF value, 37 % on average, than other latent neural states with VAF of FA was 0.42 %, VAF of GPFA 0.27 %, VAF of LFADS 0.18 % and VAF of PSID 0.29 % ($p < 0.01$, Kruskal-Wallis test and a post hoc analysis with Bonferroni correction; see Fig. S7 in the Supplementary Materials). These results support the expectation that KDLF may represent kinematics more specifically than other neural states that may also contain non-kinematic information. Thus, this may provide the opportunities to explore covariates-dependent neural representations allowing the linear relationship.

Notably, the proposed latent neural states appeared to provide movement-related information better when the arm movements became more complex, e.g., from 2D (monkeys C and M) to 3D tasks (monkey F). This was observed in both neural trajectories and decoding performance. We speculate that latent neural states directly related to kinematic parameters (i.e., KDLF) might embed kinematic information more clearly and become more effective than might other unsupervised latent neural states for encoding more complex movements. KDLF suggests a new way of dealing with the same latent factors of M1 population activity processed by the Gaussian process (GPFA) and dynamical systems (LFADS or PSID), with more potential to improve motor decoding.

This study proposed a novel latent neural state of neuronal population activity via supervised learning based on kinematic information with a nonlinear mapper. Thus, it seems natural to obtain results that latent neural states found by supervised learning with a nonlinear mapper that yielded more robust and better performance of decoding kinematics than those by PSID or unsupervised learning without any kinematic information. Therefore, previous latent neural states found in unsupervised ways are more helpful in investigating the intrinsic nature of neuronal population activity. In contrast, our latent neural states are more focused on extracting specific kinematic information. Yet our approach may provide a better way to generate useful latent neural states applicable to BMI.

This study did not evaluate the proposed approach in online BMIs, so clinical validations are unavailable. However, since one of the critical factors for clinical demands in BMIs is reliable decoding performance, we expect that our approach may be advantageous for clinical BMIs. Also, considering clinical applications where actual movements are often unavailable in participants with paralysis, one can design virtual movements for BMI training based on the examination of which kinematic parameters appropriately explain latent factors given daily. This will make BMI training and decoding more effective following kinematics-related neural information available daily. Finally, we would like to underscore that our approach may be extended to find specific latent neural states related to any behavioral covariates, including cognitive and sensory variables, which would offer a useful means to understand neural underpinnings of behavior on a single trial basis.

Acknowledgment

The authors would like to thank Dr. Andrew B. Schwartz at the Department of Neurobiology, University of Pittsburgh, Pittsburgh, PA, USA, for allowing us to use of dataset (monkey F) for this study.

References

[1] J. E. Downey, N. Schwed, S. M. Chase, A. B. Schwartz, and J. L. Collinger, "Intracortical recording stability in human brain–computer interface users," *J. Neural Eng.*, vol. 15, no. 4, Aug. 2018, Art. no. 046016, doi: 10.1088/1741-2552/aab7a0.

[2] B. Wodlinger, J. E. Downey, E. C. Tyler-Kabara, A. B. Schwartz, M. L. Boninger, and J. L. Collinger, "Ten-dimensional anthropomorphic arm control in a human brain–machine interface: Difficulties, solutions, and limitations," *J. Neural Eng.*, vol. 12, no. 1, Feb. 2015, Art. no. 016011, doi: 10.1088/1741-2560/12/1/016011.

[3] J. L. Collinger *et al.*, "Collaborative approach in the development of high-performance brain-computer interfaces for a neuroprosthetic arm: Translation from animal models to human control," *Clin. Transl. Sci.*, vol. 7, no. 1, pp. 52–59, Feb. 2014, doi: 10.1111/cts.12086.

[4] J. Dushanova and J. Donoghue, "Neurons in primary motor cortex engaged during action observation," *Eur. J. Neurosci.*, vol. 31, no. 2, pp. 386–398, Jan. 2010, doi: 10.1111/j.1460-9568.2009.07067.x.

[5] J. D. Simeral, S.-P. Kim, M. J. Black, J. P. Donoghue, and L. R. Hochberg, "Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array," *J. Neural Eng.*, vol. 8, no. 2, Apr. 2011, Art. no. 025027, doi: 10.1088/1741-2560/8/2/025027.

[6] J. L. Collinger, R. A. Gaunt, and A. B. Schwartz, "Progress towards restoring upper limb movement and sensation through intracortical brain-computer interfaces," *Current Opinion Biomed. Eng.*, vol. 8, pp. 84–92, Dec. 2018, doi: 10.1016/j.cobme.2018.11.005.

[7] J. M. Carmena *et al.*, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biol.*, vol. 1, no. 2, pp. 193–208, 2003, doi: 10.1371/journal.pbio.0000042.

[8] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal," *Nature*, vol. 416, pp. 141–142, Mar. 2002, doi: 10.1038/416141a.

[9] M. M. Churchland, A. Afshar, and K. V. Shenoy, "A central source of movement variability," *Neuron*, vol. 52, no. 6, pp. 1085–1096, Dec. 2006, doi: 10.1016/j.neuron.2006.10.034.

[10] E. E. Fetz, "Are movement parameters recognizably coded in the activity of single neurons?" *Behav. Brain Sci.*, vol. 15, no. 4, pp. 679–690, 1992.

[11] C. Pandarinath *et al.*, "Inferring single-trial neural population dynamics using sequential auto-encoders," *Nature Methods*, vol. 15, no. 10, pp. 805–815, Oct. 2018, doi: 10.1038/s41592-018-0109-9.

[12] G. Santhanam *et al.*, "Factor-analysis methods for higher-performance neural prostheses," *J. Neurophysiol.*, vol. 102, no. 2, pp. 1315–1330, 2009, doi: 10.1152/jn.00097.2009.

[13] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity," *J. Neurophysiol.*, vol. 102, no. 3, pp. 614–635, 2009, doi: 10.1152/jn.90941.2008.

[14] J. C. Kao *et al.*, "Single-trial dynamics of motor cortex and their applications to brain-machine interfaces," *Nature Commun.*, vol. 6, no. 1, Nov. 2015, Art. no. 7759, doi: 10.1038/ncomms8759.

[15] M. M. Churchland *et al.*, "Neural population dynamics during reaching," *Nature*, vol. 487, no. 7405, pp. 51–56, 2012, doi: 10.1038/nature11129.

[16] J. P. Cunningham and B. M. Yu, "Dimensionality reduction for large-scale neural recordings," *Nature Neurosci.*, vol. 17, no. 11, pp. 1500–1509, 2014, doi: 10.1038/nn.3776.

[17] J. A. Gallego, M. G. Perich, L. E. Miller, and S. A. Solla, "Neural manifolds for the control of movement," *Neuron*, vol. 94, no. 5, pp. 978–984, 2017, doi: 10.1016/j.neuron.2017.05.025.

[18] C. Pandarinath *et al.*, "Latent factors and dynamics in motor cortex and their application to brain–machine interfaces," *J. Neurosci.*, vol. 38, no. 44, pp. 9390–9401, Oct. 2018, doi: 10.1523/JNEUROSCI.1669-18.2018.

[19] M. G. Perich *et al.*, "Motor cortical dynamics are shaped by multiple distinct subspaces during naturalistic behavior," *BioRxiv*, vol. 2020, pp. 1–19, Oct. 2020, doi: doi.org/10.1101/2020.07.30.228767.

[20] K. C. Ames, S. I. Ryu, and K. V. Shenoy, "Neural dynamics of reaching following incorrect or absent motor preparation," *Neuron*, vol. 81, no. 2, pp. 438–451, Jan. 2014, doi: 10.1016/j.neuron.2013.11.003.

[21] O. G. Sani, H. Abbaspourazad, Y. T. Wong, B. Pesaran, and M. M. Shanechi, "Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification," *Nature Neurosci.*, vol. 24, no. 1, pp. 140–149, Jan. 2021, doi: 10.1038/s41593-020-00733-0.

[22] P. T. Sadtler *et al.*, "Neural constraints on learning," *Nature*, vol. 512, no. 7515, pp. 423–426, 2014, doi: 10.1038/nature13665.

[23] M. T. Kaufman, M. M. Churchland, S. I. Ryu, and K. V. Shenoy, "Cortical activity in the null space: Permitting preparation without movement," *Nature Neurosci.*, vol. 17, no. 3, pp. 440–448, Mar. 2014, doi: 10.1038/nn.3643.

[24] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[25] N. Ahmadi, T. Constandinou, and C.-S. Bouganis, "Improved spike-based brain-machine interface using Bayesian adaptive kernel smoother and deep learning," *TechRxiv. Prepr.*, vol. 2020, pp. 1–11, Oct. 2020, doi: 10.36227/techrxiv.12383600.v1.

[26] G. Andrew, R. Aroral, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. 30th Int. Conf. Mach. Learn.*, vol. 28, 2013, pp. 1247–1255.

[27] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "On deep multi-view representation learning," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 2, 2015, pp. 1083–1092.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[29] M.-K. Kim, J.-W. Sohn, and S.-P. Kim, "Decoding kinematic information from primary motor cortex ensemble activities using a deep canonical correlation analysis," *Frontiers Neurosci.*, vol. 14, pp. 1–17, Oct. 2020, doi: 10.3389/fnins.2020.509364.

[30] R. D. Flint, E. W. Lindberg, L. R. Jordan, L. E. Miller, and M. W. Slutzky, "Accurate decoding of reaching movements from field potentials in the absence of spikes," *J. Neural Eng.*, vol. 9, no. 4, Aug. 2012, Art. no. 046006, doi: 10.1088/1741-2560/9/4/046006.

[31] P. N. Lawlor, M. G. Perich, L. E. Miller, and K. P. Kording, "Linear-nonlinear-time-warp-Poisson models of neural activity," *J. Comput. Neurosci.*, vol. 45, no. 3, pp. 173–191, Dec. 2018, doi: 10.1007/s10827-018-0696-6.

[32] M. G. Perich, P. N. Lawlor, K. P. Kording, and L. E. Miller, "Extracellular neural recordings from macaque primary and dorsal premotor motor cortex during a sequential reaching task," CRCNS.org, 2018, doi: 10.6080/K0FT8J72.

[33] S. B. Suway *et al.*, "Temporally segmented directionality in the motor cortex," *Cerebral Cortex*, vol. 28, no. 7, pp. 2326–2339, Jul. 2018, doi: 10.1093/cercor/bhx133.

[34] M. Velliste, S. D. Kennedy, A. B. Schwartz, A. S. Whitford, J.-W. Sohn, and A. J. C. McMorland, "Motor cortical correlates of arm resting in the context of a reaching task and implications for prosthetic control," *J. Neurosci.*, vol. 34, no. 17, pp. 6011–6022, 2014, doi: 10.1523/JNEUROSCI.3520-13.2014.

[35] M. A. Gelbart, J. Snoek, and R. Adams, "Bayesian optimization with unknown constraints," in *Proc. 13th Conf. Uncertainty Artif. Intell.*, 2014, pp. 250–259, doi: 10.5555/3020751.3020778.

[36] D. W. Moran and A. B. Schwartz, "Motor cortical representation of speed and direction during reaching," *J. Neurophysiol.*, vol. 82, no. 5, pp. 2676–2692, Nov. 1999,doi: 10.1152/jn.1999.82.5.2676.

[37] J. A. Hennig *et al.*, "Learning is shaped by abrupt changes in neural engagement," *Nature Neurosci.*, vol. 24, no. 5, pp. 727–736, May 2021, doi: 10.1038/s41593-021-00822-8.

[38] A. D. Degenhart *et al.*, "Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity," *Nature Biomed. Eng.*, vol. 4, no. 7, pp. 672–685, Jul. 2020, doi: 10.1038/s41551-020-0542-9.

[39] B. M. Yu, A. Afshar, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, "Extracting dynamical structure embedded in neural activity," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1545–1552.

[40] B. Petreska *et al.*, "Dynamical segmentation of single trials from population neural data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1–9.