

ARTICLE

Received 3 Feb 2016 | Accepted 29 Oct 2016 | Published 13 Dec 2016

DOI: 10.1038/ncomms13749

OPEN

# Making brain-machine interfaces robust to future neural variability

David Sussillo<sup>1,2,\*</sup>, Sergey D. Stavisky<sup>3,\*</sup>, Jonathan C. Kao<sup>1,\*</sup>, Stephen I. Ryu<sup>1,4</sup> & Krishna V. Shenoy<sup>1,2,3,5,6,7</sup>

A major hurdle to clinical translation of brain-machine interfaces (BMIs) is that current decoders, which are trained from a small quantity of recent data, become ineffective when neural recording conditions subsequently change. We tested whether a decoder could be made more robust to future neural variability by training it to handle a variety of recording conditions sampled from months of previously collected data as well as synthetic training data perturbations. We developed a new multiplicative recurrent neural network BMI decoder that successfully learned a large variety of neural-to-kinematic mappings and became more robust with larger training data sets. Here we demonstrate that when tested with a non-human primate preclinical BMI model, this decoder is robust under conditions that disabled a state-of-the-art Kalman filter-based decoder. These results validate a new BMI strategy in which accumulated data history are effectively harnessed, and may facilitate reliable BMI use by reducing decoder retraining downtime.

<sup>1</sup>Electrical Engineering Department, Stanford University, Stanford, California 94305, USA. <sup>2</sup>Stanford Neurosciences Institute, Stanford, California 94305, USA. <sup>3</sup>Neurosciences Graduate Program, Stanford, California 94305, USA. <sup>4</sup>Palo Alto Medical Foundation, Palo Alto, California 94301, USA. <sup>5</sup>Neurobiology and Bioengineering Departments, Stanford, California 94305, USA. <sup>6</sup>Bio-X Program, Stanford, California 94305, USA. <sup>7</sup>Howard Hughes Medical Institute at Stanford University Stanford University, Stanford, 94305 California, USA. \* These authors contributed equally to this work. Correspondence and requests for materials should be addressed to K.V.S. (email: shenoy@stanford.edu).

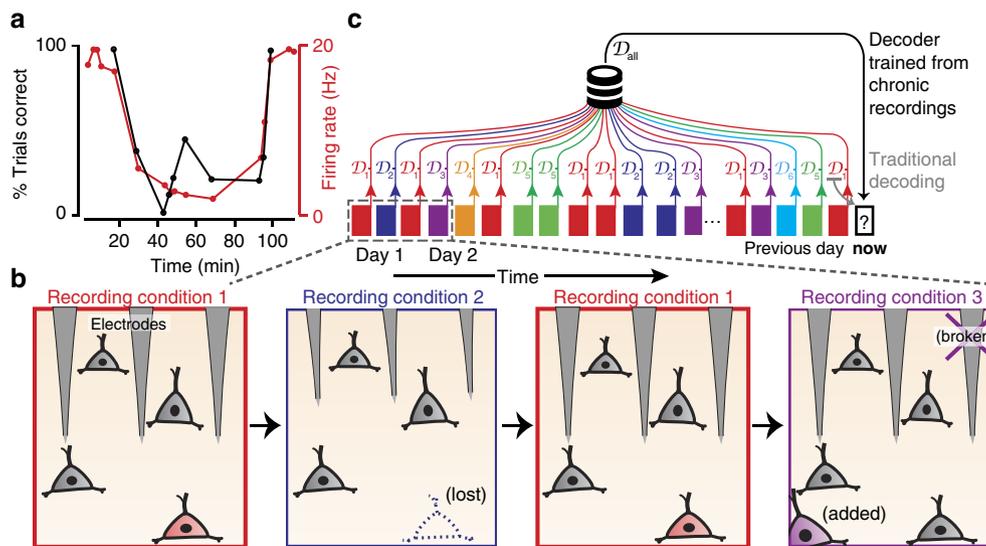
**B**rain-machine interfaces (BMIs) can restore motor function and communication to people with paralysis<sup>1,2</sup>. Progress has been particularly strong towards enabling two-dimensional (2D) computer cursor control, which may allow versatile communications prostheses<sup>3–5</sup>. Cursor-control performance has approached that of the native hand in recent macaque studies<sup>6,7</sup>, but this was done under favourable laboratory conditions where neural recordings are often stable both during and across BMI sessions<sup>8–11</sup>. In contrast to these preclinical studies, one of the major challenges impeding BMI use by human clinical trial participants is the high degree of within- and across-day variability in neural recording conditions (Fig. 1a)<sup>12–16</sup>. We use the term ‘recording condition’ to broadly encompass the combination of factors that together determine the relationship between observed neural activity and intended kinematics. These factors include the relative position of the electrodes and surrounding neurons (diagrammed in Fig. 1b), variability in sensor properties such as impedance or wiring quality, noise sources and biological factors such as cognitive state or medications. Existing neural decoding algorithms are poorly suited to handle variability in recording condition, resulting in intermittent performance and a need for frequent decoder retraining<sup>4,5,13,17</sup>.

The clinical viability of BMIs would be much improved by making decoders robust to recording condition changes<sup>18,19</sup>, and several recent studies have focused on this problem (for example, refs 4,10,20–29). We can broadly divide the conditions that a BMI will encounter into one of two types: (1) conditions that are completely different from what has been previously encountered; and (2) conditions that share some commonality with ones previously encountered. For existing BMI methods, both of these

situations necessitate some interruption of function while the decoder is updated to handle the new condition. One strategy for minimizing this interruption is to use adaptive decoders, which update their parameters based on new data collected during the BMI’s use (rather than collecting new training data for a *de novo* decoder) to try to better match the new recording condition<sup>4,10,20–29</sup>. In the first case, this is likely the best that can be done. But in the second case, BMI interruption could in principle be avoided altogether by a decoder capable of exploiting the similarities between the current and previously encountered conditions (Fig. 1c).

We were motivated to try this complimentary strategy because chronic BMI systems do typically encounter recording conditions in which there is some commonality with past recording conditions<sup>8,10,13,14,27,28,30–32</sup>. Furthermore, these systems generate and store months, or even years, of neural and kinematic data as part of their routine use. Almost all of these past data are left unused in existing BMI systems: decoders are trained using the most recently available data, typically from a block of calibration trials at the start of that day’s experiment, or from a recent previous experiment<sup>33</sup>. Using this historical data would be difficult for most BMI decoders, as they are linear (for example, refs 2,6). Linear decoders are prone to underfitting heterogeneous training sets, such as those that might be sampled from months of data. To overcome this limitation, an essential aspect of our approach is to use a nonlinear and computationally ‘powerful’ decoder (that is, one capable of approximating any complex, nonlinear dynamical system), which should be capable of learning a diverse set of neural-to-kinematic mappings.

Specifically, we tested whether one could gain traction on the decoder robustness problem by exploiting this idle wealth of



**Figure 1 | Strategy for training a decoder robust to recording condition changes.** (a) Example data from a BMI clinical trial showing sudden decoder failure caused by a recording condition change. The black trace shows the participant’s closed-loop performance over the course of an experiment using a fixed Kalman filter. An abrupt drop in performance coincides with a reduction in the observed firing rate (red trace) of a neuron with a high decoder weight. Both the neuron’s firing rate and decoder performance spontaneously recover ~40 min later. Adapted from Figure 7 of ref. 13. (b) A cartoon depicting one hypothetical cause of the aforementioned change: micro-motion of the electrodes leads to Recording Condition 2, in which spikes from the red-shaded neuron are lost. BMI recovery corresponds to a shift back to Condition 1. Over time, further changes will result in additional recording conditions; for example, Condition 3 is shown caused by a disconnected electrode and an additional neuron entering recording range. (c) Recording conditions (schematized by the coloured rectangles) will vary over the course of chronic intracortical BMI use. We hypothesize that oftentimes new conditions are similar to ones previously encountered (repeated colours). Typically, decoders are fit from short blocks of training data and are only effective under that recording condition (decoders  $D_1, D_2, \dots$ ). Consider the goal of training a decoder for use at time ‘now’ (black rectangle on right). Standard practice is to use decoder  $D_1$  trained from the most recently available data (for example, from the previous day or the start of the current experiment).  $D_1$  will perform poorly if the recording condition encountered differs from its training data. To increase the likelihood of having a decoder that will perform well given the current recording condition, we tested a new class of decoder,  $D_{all}$ , trained using a large collection of previous recording conditions.

stored data using an artificial recurrent neural network (RNN). We did this with a three-pronged approach. The first was the use of the nonlinear RNN. The second was to train the decoder from many months of previously recorded data. Third, to ‘harden’ the decoder against being too reliant on any given pattern of inputs, we artificially injected additional variability into the data during decoder training.

The fact that conventional state-of-the-art decoding methods, which tend to be linear or at least of limited computational complexity<sup>34</sup>, work well for closed-loop BMI control of 2D cursors demonstrates that the model mismatch of assuming linear neural-to-kinematic mappings is well tolerated for a given recording condition. Nevertheless, when neural-to-kinematic mappings change over time, a conventional decoder trained on many days’ data is almost certainly not going to fully benefit from this abundance of the data. This is because it requires a nonlinear algorithm to learn a set of different context-dependent mappings, even if these individual mappings from neural firing rates to kinematics were entirely linear (which they are not). Methods such as linear Kalman filters can at best only learn an average mapping, ‘splitting the difference’ to reduce error across days in the training set. This approach is not well-suited for most of the recording conditions. We therefore developed a new BMI decoder using a nonlinear RNN variant called the multiplicative recurrent neural network (MRNN) developed by Sutskever and colleagues<sup>35</sup> using their Hessian-free technique for training RNNs<sup>36</sup>. Several properties of the MRNN architecture, which was originally used for character-level language modelling, make it attractive for this neural prosthetic application. First, it is recurrent, and can therefore ‘remember’ state across time (for example, during the course of a movement), potentially better matching the time-varying, complex relationships between neural firing rates and kinematics<sup>37,38</sup>. Second, its ‘multiplicative’ architecture increases computational power by allowing the neural inputs to influence the internal dynamics of the RNN by changing the recurrent weights (Fig. 2a). Loosely speaking, this allows the MRNN to learn a ‘library’ of different neural-to-kinematic mappings that are appropriate to different recording conditions. The MRNN was our specific choice of nonlinear method for learning a variety of neural-to-kinematic mappings, but this general approach is likely to work well with many out-of-the-box RNN variants, such as a standard RNN (for example, ref. 38) or LSTM<sup>39</sup>. Our approach is also completely complementary to adaptive decoding.

We evaluated decoders using two non-human primates implanted with chronic multielectrode arrays similar to those used in ongoing clinical trials. We first show that training the MRNN with more data from previous recording sessions improves accuracy when decoding new neural data, and that a single MRNN can be trained to accurately decode hand reach velocities across hundreds of days. We next present closed-loop results showing that an MRNN trained with many days’ worth of data is much more robust than a state-of-the-art Kalman filter-based decoder (the Feedback Intention Trained Kalman filter, or FIT-KF<sup>40</sup>) to two types of recording condition changes likely to be encountered in clinical BMI use: the unexpected loss of signals from highly-informative electrodes, and day-to-day changes. Finally, we show that this robustness does not come at the cost of reduced performance under more ideal (unperturbed) conditions: in the absence of artificial challenges, the MRNN provides excellent closed-loop BMI performance and slightly outperforms the FIT-KF. To our knowledge, this is the first attempt to improve robustness by using a large and heterogeneous training dataset: we used roughly two orders of magnitude more data than in previous closed-loop studies.

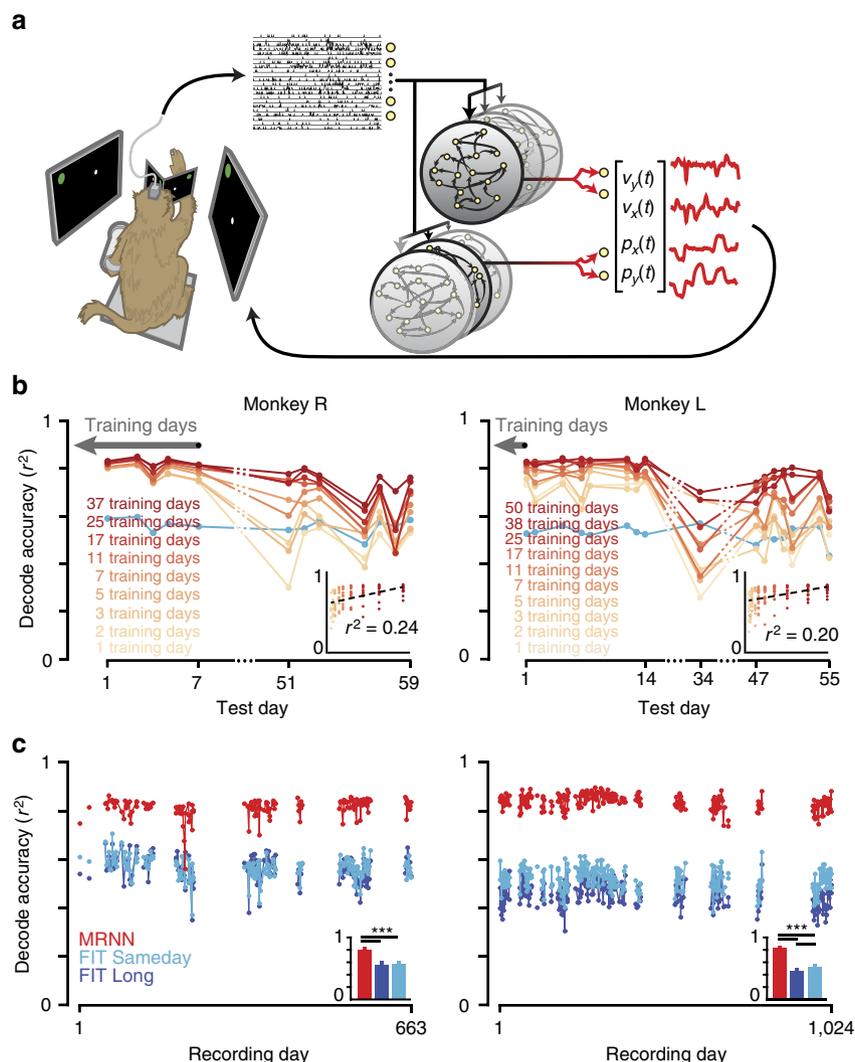
## Results

**MRNN performance improves with more data.** We first tested whether training the MRNN with many days’ worth of data can improve offline decoder performance across a range of recording conditions. This strategy was motivated by our observation that the neural correlates of reaching—as recorded with chronic arrays—showed day-to-day similarities (Supplementary Fig. 1). For a typical recording session, the most similar recording came from a chronologically close day, but occasionally the most similar recording condition was found in chronologically distant data. MRNN decoders were able to exploit these similarities: Figure 2b shows that as more days’ data (each consisting of ~500 point to point reaches) were used to train the decoder, the accuracy of reconstructing reach velocities, measured as the square of the Pearson’s correlation coefficient between true and decoded test data set velocity, increased (positive correlation between number of training days and decoded velocity accuracy,  $r^2 = 0.24$ ,  $P = 2.3e - 7$  for monkey R ( $n = 99$ ),  $r^2 = 0.20$ ,  $P = 3.2e - 9$  for monkey L ( $n = 160$ ), linear regression). In particular, these results show that using more training data substantially increased the decode accuracy for the ‘hard’ days that challenged decoders trained with only a few days’ data (for example, test day 51 for monkey R). Further, this improvement did not come at the cost of worse performance on the initially ‘easy’ test days. These results demonstrate that larger training data sets better prepare the MRNN for a variety of recording conditions, and that learning to decode additional recording conditions did not diminish the MRNN’s capability to reconstruct kinematics under recording conditions that it had already ‘mastered’. There was not a performance versus robustness trade-off.

We then tested whether the MRNN’s computational capacity could be pushed even further by training it using the data from 154 (250) different days’ recording sessions from monkey R (L), which spanned 22 (34) months (Fig. 2c). The MRNN’s offline decode accuracy was  $r^2 = 0.81 \pm 0.04$  (mean  $\pm$  s.d., monkey R) and  $r^2 = 0.84 \pm 0.03$  (monkey L) across all these recording sessions’ held-out test trials. For comparison, we tested the decode accuracy of the FIT-KF trained in two ways: either specifically using reaching data from that particular day (‘FIT Sameday’), or trained on the same large multiday training data set (‘FIT Long’). Despite the multitude of recording conditions that the MRNN had to learn, on every test day each monkey’s single MRNN outperformed that day’s FIT Sameday filter (monkey R ( $n = 154$  samples): FIT Sameday  $r^2 = 0.57 \pm 0.05$ ,  $P = 1.2e - 153$  signed-rank test comparing all days’ FIT Sameday and MRNN  $r^2$ ; monkey L ( $n = 250$  samples):  $r^2 = 0.52 \pm 0.05$ ,  $P = 2.1e - 319$ ). Unsurprisingly, a linear FIT-KF did not benefit from being trained with the same large multiday training set and also performed worse than the MRNN (monkey R: FIT Long  $r^2 = 0.56$ ,  $P = 5.1e - 27$  comparing all days’ FIT Long to MRNN  $r^2$ ; monkey L:  $r^2 = 0.46 \pm 0.05$ ,  $P = 9.3e - 43$ ).

While these offline results demonstrate that the MRNN can learn a variety of recording conditions, experiments are required to evaluate whether this type of training leads to increased decoder robustness under closed-loop BMI cursor control. In closed-loop use, the BMI user updates his or her motor commands as a result of visual feedback, resulting in distributions of neural activity that are different than that of the training set. Thus, results from offline simulation and closed-loop BMI control may differ<sup>32,41–43</sup>. To this end, we next report closed-loop experiments that demonstrate the benefit of this training approach.

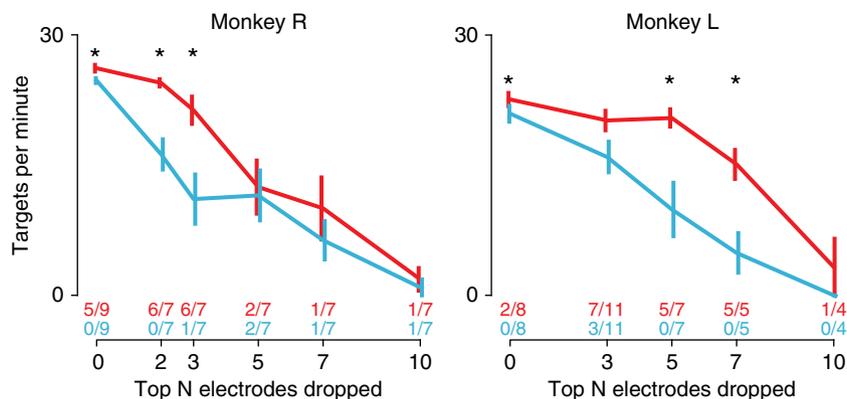
**Robustness to unexpected loss of informative electrodes.** We next performed closed-loop BMI cursor-control experiments



**Figure 2 | An MRNN decoder can harness large training data sets.** (a) A monkey performed a target acquisition task using his hand while multiunit spikes were recorded from multielectrode arrays in motor cortex. Data from many days were used to train two MRNNs such that velocity and position were read out from the state of their respective internal dynamics. These MRNN internal dynamics are a function of the binned neural spike counts; thus, the MRNN can conceptually be thought of as selecting an appropriate decoder at any given time based on the neural activity. (b) We evaluated each MRNN's ability to reconstruct offline hand velocity on 12 (16) monkey R (L) test days after training with increasing numbers of previous days' data sets. Training data were added by looking further back in time so as to not conflate training data recency with data corpus size. In monkey R, early test days also contributed training data (with test trials held out). In monkey L, from whom more suitable data was available, the training data sets started with the day prior to the first test day. More training data (darker coloured traces) improved decode accuracy, especially when decoding more chronologically distant recording conditions. We also plotted performance of a FIT Kalman filter trained from each individual day's training data ('FIT Sameday', light blue). (Insets) show the same MRNN data in a scatter plot of decode accuracy versus number of training days (99 data points for monkey R, 160 for L). Linear fit trend lines reveal a significant positive correlation. (c) An MRNN (red trace) was trained with data from 154 (250) monkey R (L) recording days spanning many months. Its offline decoding accuracy on held-out trials from each of these same days was compared with that of the FIT Sameday (light blue). We also tested a single FIT-KF trained using the same large dataset as the MRNN ('FIT Long', dark blue). Gaps in the connecting lines denote recording gaps of more than ten days. (Insets) mean  $\pm$  s.d. decode accuracy across all recording days. Stars denote  $P < 0.001$  differences (signed-rank test). The MRNN outperformed both types of FIT-KF decoders on every day's dataset.

to test the MRNN's robustness to recording condition changes. The first set of experiments challenged the decoder with an unexpected loss of inputs from multiple electrodes. The MRNN was trained with a large corpus of hand-reaching training data up through the previous day's session (119–129 training days for monkey R, 212–230 days for monkey L). Then, its closed-loop performance was evaluated on a Radial 8 Task, while the selected electrodes' input firing rates were artificially set to zero. By changing how many of the most informative electrodes were dropped ('informative' as determined by their mutual information with reach direction; see Methods), we could systematically

vary the severity of the challenge. Since this experiment was meant to simulate sudden failure of electrodes during BMI use (after the decoder had already been trained), we did not retrain or otherwise modify the decoder based on knowledge of which electrodes were dropped. There were no prior instances of these dropped electrode sets having zero firing rates in the repository of previously collected training data (Supplementary Fig. 2). Thus, this scenario is an example of an unfamiliar recording condition (zero firing rates on the dropped electrodes) having commonality with a previously encountered condition (the patterns of activity on the remaining electrodes).



**Figure 3 | Robustness to unexpected loss of the most important electrodes.** Closed-loop BMI performance using the MRNN (red) and FIT Sameday (blue) decoders while simulating an unexpected loss of up to 10 electrodes by setting the firing rates of these electrodes to zero. The mean and s.e.m. across experimental sessions' targets per minute performance is shown for each decoder as a function of how many electrodes were removed. Stars denote conditions for which the MRNN significantly outperformed FIT Sameday across sessions ( $P < 0.05$ , paired  $t$ -test). The text above each condition's horizontal axis tick specifies for how many of the individual evaluation days MRNN (red fraction) or FIT Sameday (blue fraction) performed significantly better than the other decoder according to single-session metrics of success rate and time to target. Electrode-dropping order was determined by the mutual information between that electrode's spike count and target direction during arm-controlled reaches.

We found that the MRNN was robust to severe electrode-dropping challenges. It suffered only a modest loss of performance after losing up to the best 3 (monkey R) or 5 (monkey L) electrodes (Fig. 3). We compared this with the electrode-dropped performance of a FIT-KF decoder trained with hand-reaching calibration data from the beginning of that day's experiment<sup>6,40</sup> ('FIT Sameday') by alternating blocks of MRNN and FIT Sameday control in an 'AB AB' interleaved experiment design. FIT Sameday decoder's performance worsened markedly when faced with this challenge. Across all electrode-dropped conditions, Monkey R acquired 52% more targets per minute using the MRNN, while Monkey L acquired 92% more targets. Supplementary Movie 2 shows a side-by-side comparison of the MRNN and FIT Sameday decoders with the three most informative electrodes dropped.

Although the past data sets used to train the MRNN never had these specific sets of highly important electrodes disabled, our technique of artificially perturbing the true neural activity during MRNN training did generate training examples with reduced firing rates on various electrodes (as well as examples with increased firing rates). The MRNN had therefore been broadly trained to be robust to firing rate reduction on subsets of its inputs. Subsequent closed-loop comparisons of MRNN electrode-dropping performance with and without this training data augmentation confirmed its importance (Supplementary Fig. 3a). An additional offline decoding simulation, in which MRNN decoders were trained with varying data set sizes with and without training data augmentation, further shows that both the MRNN architecture and its training data augmentation are important for robustness to electrode dropping (Supplementary Fig. 4). These analyses also suggest that when data augmentation is used, large training data set size does not impart additional robustness to these particular recording condition changes. This is not surprising given that the previous data sets did not include examples of these electrodes being dropped.

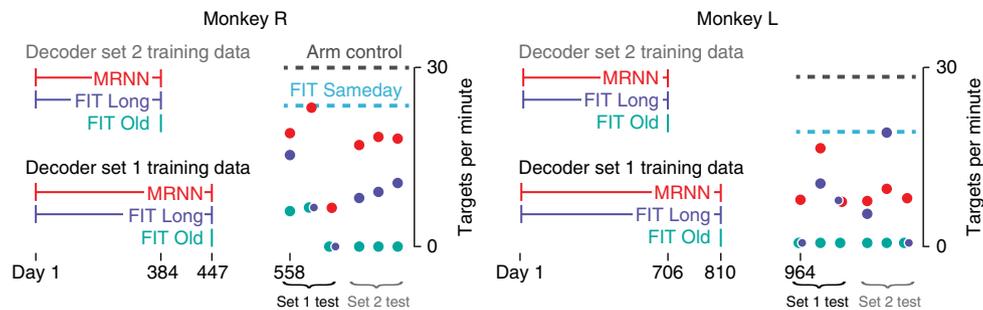
#### Robustness to naturally sampled recording condition changes.

The second set of closed-loop robustness experiments challenged the MRNN with naturally occurring day-to-day recording condition changes. In contrast to the highly variable recording conditions encountered in human BMI clinical trials, neural recordings in our laboratory set-up are stable within a day and

typically quite stable on the time scale of days (Supplementary Fig. 2; ref. 10). Therefore, to challenge the MRNN and FIT-KF decoders with greater recording condition variability, we evaluated them after withholding the most recent several months of recordings from the training data. We refer to this many-month interval between the most recent training data day and the first test day as the training data 'gap' in these 'stale training data' experiments. The gaps were chosen arbitrarily within the available data, but to reduce the chance of outlier results, we repeated the experiment with two different gaps for each monkey.

For each gap, we trained the MRNN with a large data set consisting of many months of recordings preceding the gap and compared it with two different types of FIT-KF decoders. The 'FIT Old' decoder was trained from the most recent available training day (that is, the day immediately preceding the gap); this approach was motivated under the assumption that the most recent data were most likely to be similar to the current day's recording condition. The 'FIT Long' decoder was trained from the same multiday data set used to train the MRNN and served as a comparison in which a conventional decoder is provided with the same quantity of data as the MRNN. The logic underlying this FIT Long approach is that despite the Kalman filter being ill-suited for fitting multiple heterogeneous data sets, this 'averaged' decoder might still perform better than the FIT Old trained using a single distant day.

We found that the MRNN was the only decoder that was reliably usable when trained with stale data (Fig. 4). FIT Old performed very poorly in both monkeys, failing completely (defined as the monkey being unable to complete a block using the decoder, see Methods) in 4/6 monkey R experimental sessions and 6/6 monkey L sessions. FIT Long performed better than FIT Old, but its performance was highly variable—it was usable on some test days but failed on others. In Monkey R, the across-days average acquisition rate was 105% higher for the MRNN than FIT Long ( $P = 4.9e - 4$ , paired  $t$ -test). Monkey L's MRNN did not perform as consistently well as Monkey R's, but nevertheless demonstrated a trend of outperforming FIT Long (32% improvement,  $P = 0.45$ ), in addition to decidedly outperforming FIT Old, which failed every session. Although monkey L's FIT Long outperformed the MRNN on one test day, on all other test days FIT Long was either similar to, or substantially worse than, MRNN. Moreover, whereas the MRNN could be used to

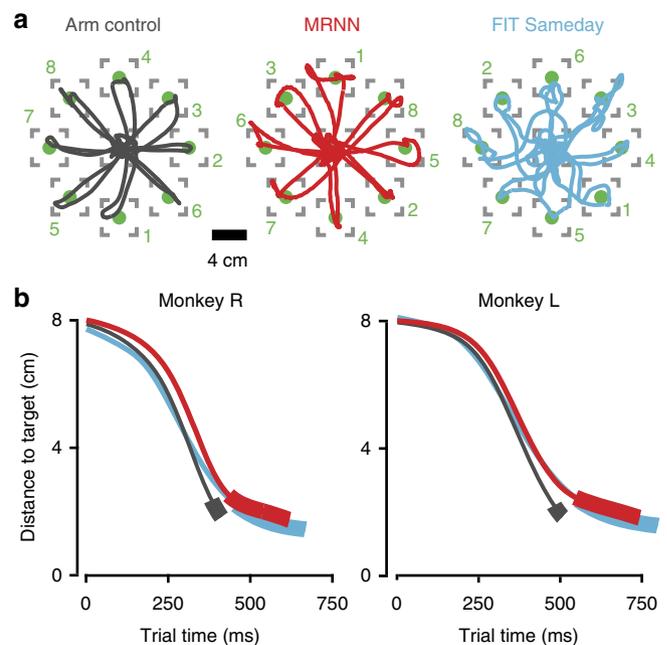


**Figure 4 | Robustness to naturally occurring recording condition changes.** We created decoder evaluation conditions in which the neural inputs were likely to be different from much of the training data by withholding access to the most recent several months of data. Each circle corresponds to the mean closed-loop BMI performance using these ‘stale’ MRNN (red), FIT Long (dark blue) and FIT Old (teal) decoders when evaluated on six different experiment days spanning 7 (13) days in monkey R (L). Each test day, these three decoders, as well as a FIT Sameday decoder trained from that day’s arm reaches, were evaluated in an interleaved block design. The legend bars also denote the time periods from which training data for each stale decoder came from. We repeated the experiments for a second set of decoders to reduce the chance that the results were particular to the specific training data gap chosen. The training data periods contained 82 and 92 data sets (monkey R) and 189 and 200 training data sets (monkey L). The only decoder that was consistently usable, that is, did not fail on any test days, was the MRNN. To aid the interpretation of these stale decoder performances, we show the average performance across the six experiment days using arm control (grey dashed line) or a FIT Sameday decoder (blue dashed line).

control the cursor every day, FIT Long was not even capable of acquiring targets on some days. Further tests of additional FIT Old decoders confirmed that they generally perform poorly (Supplementary Fig. 5). The lack of consistent usability by any of the FIT-KF decoders (Old or Long) demonstrates that having access to a large repository of stale training data does not enable training a single Kalman filter that is robust to day-to-day variability in recording conditions. In contrast, an MRNN trained with this large data set was consistently usable.

To further demonstrate the consistency of these results, we performed offline simulations in which we tested MRNN decoders on additional sets of training and test data sets separated by a gap. Each set was non-overlapping with the others, and together they spanned a wide range of each animal’s research career. We observed the same trends in these offline simulations: MRNNs trained with many previous days of training data outperformed FIT Old and FIT Long decoders (Supplementary Fig. 6). In these analyses, we also dissected which components of our decoding strategy contributed to the MRNN’s robustness. We did this by comparing MRNNs trained with varying numbers of days preceding the gap, with or without training data spike rate perturbations. The results show that training using more data, and to a lesser extent incorporating data augmentation (see also closed-loop comparisons in Supplementary Fig. 3b), contributed to the MRNN’s robustness to naturally occurring recording condition changes.

**High-performance BMI using the MRNN decoder.** Finally, we note that the MRNN’s robustness to challenging recording conditions did not come at the cost of reduced performance under more ‘ideal’ conditions, that is, without electrode dropping or stale training data. During the electrode-dropping experiments, we also evaluated the MRNN’s closed-loop performance after being trained using several months’ data up through the previous day. In this scenario, the MRNN enabled both monkeys to accurately and quickly control the cursor. Supplementary Movie 1 shows example cursor control using the MRNN. These data also allowed us to compare the MRNN’s performance with that of a FIT Sameday decoder in back-to-back ‘AB AB’ tests. Figure 5a shows representative cursor trajectories using each decoder, as well as under hand control. Figure 5b shows that across 9 experimental sessions and 4,000+ trials with each



**Figure 5 | MRNN achieves high-performance under ‘ideal’ conditions.** (a) We compared cursor control using the MRNN (red) trained from many data sets up through the previous day to the FIT Sameday (blue) trained from data collected earlier the same day, without any artificial challenges (that is, no electrodes dropped or stale training data). Cursor trajectories are shown for eight representative and consecutive centre-out-and-back trials of the Radial 8 Task. Grey boxes show the target acquisition area boundaries, and the order of target presentation is denoted with green numbers. For comparison, cursor trajectories under arm control are shown in grey. From dataset R.2014.04.03. (b) Mean distance to target, across all Radial 8 Task trials under these favourable conditions, as a function of trial time using each cursor-control mode. Thickened portions of each trace correspond to ‘dial-in time’, that is, the mean time between the first target acquisition and the final target acquisition. These MRNN and FIT Sameday data correspond to the drop 0 electrodes condition data in Fig. 3, and include 4,094 (3,278) MRNN trials and 4119 (3,305) FIT Sameday trials over 9 (8) experimental days in Monkey R (L).

decoder, Monkey R acquired targets 7.3% faster with the MRNN ( $0.619 \pm 0.324$  s mean  $\pm$  s.d. vs.  $0.668 \pm 0.469$  s,  $P = 4.2e - 6$ , rank-sum test). Monkey L acquired targets 10.8% faster with the MRNN ( $0.743 \pm 0.390$  s versus  $0.833 \pm 0.532$  s,  $P = 1.5e - 3$ , rank-sum test) across 8 sessions and 2,500 + trials using each decoder. These online results corroborate the offline results presented in Fig. 2c; both show that an MRNN trained from many days' recording conditions outperforms the FIT Kalman filter trained from training data collected at the start of the experimental session.

A potential risk inherent to a computationally powerful decoder such as the MRNN is that it will overtrain to the task structure of the training data and fail to generalize to other tasks. Most of our MRNN training data were from arm reaches on a Radial 8 Task similar to the task used for evaluation (albeit with 50% further target distance). We therefore also tested whether the MRNN enabled good cursor control on the Random Target Task, in which the target could appear in any location in a  $20 \times 20$  cm workspace (Supplementary Fig. 7). Monkey R performed the Random Target Task on two experimental sessions and averaged a 99.4% success rate, with mean distance-normalized time to target of  $0.068$  s  $\text{cm}^{-1}$ . Monkey L performed one session of this task at a 100% success rate with mean normalized time to target of  $0.075$  s  $\text{cm}^{-1}$ . To provide context for these metrics, we also measured Random Target Task performance using arm control. Monkey R's arm control success rate was 100%, with  $0.055$  s  $\text{cm}^{-1}$  mean normalized time to target, during the same experimental sessions as his MRNN Random Target Task data. Monkey L's arm control success rate was 97.7%, with  $0.055$  s  $\text{cm}^{-1}$  mean normalized time to target, during one session several days following his MRNN test.

## Discussion

We developed the MRNN decoder to help address a major problem hindering the clinical translation of BMIs: once trained, decoders can be quickly rendered ineffective due to recording condition changes. A number of complementary lines of research are aimed at making BMIs more robust, including improving sensors to record from more neurons more reliably (for example, ref. 44); decoding multiunit spikes<sup>10,30,45</sup> or local field potentials<sup>31,32,46</sup> that appear to be more stable control signals than single-unit activity; and using adaptive decoders that update their parameters to follow changing neural-to-kinematic mappings<sup>4,10,20–29,47</sup>. Here we present the MRNN as a proof-of-principle of a novel approach: build a fixed decoder whose architecture allows it to be inherently robust to recording condition changes based on the assumption that novel conditions have some similarity to previously encountered conditions.

We stress that all of these approaches are complementary in several respects. For example, a decoder that is inherently more robust to neural signal changes, such as the MRNN, would still benefit from improved sensors, could operate on a mix of input signal types including single- and multiunit spikes and field potentials, and is especially well positioned to benefit from decoder adaptation. When performance degrades due to recording condition changes, both supervised<sup>10,21–23,25,27,29</sup> and unsupervised<sup>4,20,24,26</sup> adaptive decoders need a period of time in which control is at least good enough that the algorithm can eventually infer the user's intentions and use these to update its neural-to-kinematic model. Improved robustness may 'buy enough time' to allow the decoder's adaptive component to rescue performance without interrupting prosthesis use. Here we have demonstrated the MRNN's advantages over a state-of-the-art static decoder, but comparing this strategy both

against and together with adaptive decoding remains a future direction.

We demonstrated the MRNN's robustness to two types of recording condition changes. These changes were chosen because they capture key aspects of the changes that commonly challenge BMI decoders during clinical use. The stale training data experiments showed that the MRNN was usable under conditions where the passage of time would typically require recalibration of conventional decoders such as the FIT-KF. We do not mean to suggest that in a clinical setting one would want to—or would often have to—use a BMI without any training data from the immediately preceding several months. Rather, we used this experimental design to model recording condition changes that can happen on the time scale of hours in human BMI clinical trials<sup>13</sup>. Possible reasons for the greater recording condition variability observed in human participants compared with non-human primates include: more movement of the array relative to the human brain due to larger cardiovascular pulsations and epidural space; greater variability in the state of the BMI user (health, medications, fatigue and cognitive state); and more electromagnetic interference from the environment. The MRNN can take advantage of having seen the effects of these sources of variability in previously accumulated data; it can therefore be expected to become more robust over time as it builds up a 'library' of neural-to-kinematic mappings under different recording conditions.

The electrode-dropping experiments, which demonstrated the MRNN's robustness to an unexpected loss of high-importance electrodes, are important for two reasons. First, sudden loss of input signals (for example, due to a electrode connection failure<sup>48,49</sup>), is a common BMI failure mode that can be particularly disabling to conventional BMI decoders<sup>50</sup>. The MRNN demonstrates considerable progress in addressing this so-called 'errant unit' problem. Second, these results demonstrate that the MRNN trained with artificially perturbed neural data can be relatively robust even to a recording condition change that has not been encountered in past recordings.

The MRNN's robustness did not come at the cost of diminished performance under more ideal conditions. This result is nontrivial given the robustness-focused decisions that went into its design (for example perturbing the input spike trains in the training set). Instead, we found that the MRNN was excellent under favourable conditions, slightly outperforming a state-of-the-art same day trained FIT-KF decoder. Taken together, these results demonstrate that the MRNN exhibits robustness to a variety of clinically relevant recording condition changes, without sacrificing peak performance. These advances may help to reduce the onerous need for clinical BMI users to collect frequent retraining data.

One disadvantage of this class of nonlinear decoders trained from large data sets, when compared with traditional linear decoders trained on smaller data sets, is the longer training time. In the present study, which we did not optimize for fast training, this took multiple hours. This could be substantially sped up by iteratively updating the decoder with new data instead of retraining *de novo* and by leveraging faster computation available with graphics processing units, parallel computing, or custom hardware. A second disadvantage of the MRNN is that it appears to require more training data to saturate its performance (Fig. 2b) compared with conventional methods, such as FIT-KF, that are trained from calibration data collected on the same day. We do not view this as a major limitation because the motivation for using the MRNN is to take advantage of accumulated previous recordings. Nonetheless, it will be valuable to compare the present approach with other decoder architectures and training

strategies, which may yield similar performance and robustness while requiring less training data.

The MRNN decoder's robustness was due to the combination of a large training data corpus, deliberate perturbation of the training data and a computationally powerful architecture that was able to effectively learn this diverse training data. While it may seem obvious that successfully learning more training data is better, this is not necessarily true. Older data only help a decoder if some of these past recordings capture neural-to-kinematic relationships that are similar to that of the current recording condition. Our offline and closed-loop MRNN robustness results suggest that this was indeed the case for the two monkeys used in this study. While there are indications that this will also be true in human BMI studies<sup>14</sup>, validating this remains an important future question. The relevance of old data to present recording conditions also motivates a different robustness-enhancing approach: store a library of different past decoders and evaluate each to find a decoder well-suited for the current conditions (for example, ref. 10). However, since offline analyses are poor predictors of closed-loop performance<sup>32,42,45,51</sup>, this approach necessitates a potentially lengthy decoder selection process. Using a single decoder (such as the MRNN) that works across many recording conditions avoids switching-related downtime.

In addition to training with months of previous data, we improved the MRNN's robustness by intentionally perturbing the training neural data. In the present study, we applied random Gaussian firing rate scaling based on a general assumption that the decoder should be broadly robust to both global and private shifts in observed firing rates. This perturbation type proved effective, but we believe that this approach (called data augmentation in the machine learning community) can potentially be much more powerful when combined with specific modelling of recording condition changes that the experimenter wants to train robustness against. For example, data augmentation could incorporate synthetic examples of losing a particularly error-prone set of electrodes; recording changes predicted by models of array micro-movement or degradation; and perhaps even the predicted interaction between kinematics and changes in cognitive state or task context. We believe this is an important avenue for future research.

We view the success of our specific MRNN decoder implementation as a validation of the more general BMI decoder strategy of training a computationally powerful nonlinear decoder to a large quantity of data representing many different recording conditions. This past data need not have been collected explicitly for the purpose of training as was done in this study; neural data and corresponding kinematics from past closed-loop BMI use can also serve as training data<sup>4,10</sup>. It is likely that other nonlinear decoding algorithms will also benefit from this strategy, and that there are further opportunities to advance the reliability and performance of BMIs by starting to take advantage of these devices' ability to generate large quantities of data as part of their regular use.

## Methods

**Animal model and neural recordings.** All procedures and experiments were approved by the Stanford University Institutional Animal Care and Use Committee. Experiments were conducted with adult male rhesus macaques (R and L, ages 8 and 18 years, respectively), implanted with 96-electrode Utah arrays (Blackrock Microsystems Inc., Salt Lake City, UT) using standard neuro-surgical techniques. Monkeys R and L were implanted 30 months and 74 months before the primary experiments, respectively. Monkey R had two electrode arrays implanted, one in caudal dorsal premotor cortex (PMd) and the other in primary motor cortex (M1), as estimated visually from anatomical landmarks. Monkey L had one array implanted on the border of PMd and M1. Within the context of the simple point-to-point arm and BMI reach behaviour of this study, we observed qualitatively similar response properties between these motor cortical areas; this is consistent with previous reports of a gradient of increasing

preparatory activity, rather than stark qualitative differences, as one moves more rostral from M1 (refs 52–56). Therefore, and in keeping with standard BMI decoding practices<sup>6,8,10,24,38,40,46</sup>, we did not distinguish between M1 and PMd electrodes.

Behavioural control and neural decode were run on separate PCs using the xPC Target platform (Mathworks, Natick, MA), enabling millisecond-timing precision for all computations. Neural data were initially processed by Cerebus recording system(s) (Blackrock Microsystems Inc., Salt Lake City, UT) and were available to the behavioural control system within  $5 \pm 1$  ms. Spike counts were collected by applying a single negative threshold, set to  $-4.5$  times the root mean square of the spike band of each electrode. We decoded 'threshold crossings', which contain spikes from one or more neurons in the electrode's vicinity, as per standard practice for intracortical BMIs<sup>1,4,6,7,10,15,16,31,38,40</sup> because threshold crossings provide roughly comparable population-level velocity decode performance to sorted single-unit activity, without time-consuming sorting<sup>30,45,57–59</sup>, and may be more stable over time<sup>30,45</sup>. To orient the reader to the quality of the neural signals available during this study, Supplementary Note 1 provides statistics of several measures of electrodes' 'tuning' and cross-talk.

**Behavioural tasks.** We trained the monkeys to acquire targets with a virtual cursor controlled by either the position of the hand contralateral to the arrays or directly from neural activity. Reaches to virtual targets were made in a 2D frontoparallel plane presented within a 3D environment (MSMS, MDDE, USC, Los Angeles, CA) generated using a Wheatstone stereograph fused from two LCD monitors with refresh rates at 120 Hz, yielding frame updates within  $7 \pm 4$  ms (ref. 43). Hand position was measured with an infrared reflective bead tracking system at 60 Hz (Polaris, Northern Digital, Ontario, Canada). During BMI control, we allowed the monkey's reaching arm to be unrestrained<sup>47,60</sup> so as to not impose a constraint upon the monkey that during BMI control he must generate neural activity that does not produce overt movement<sup>61</sup>.

In the Radial 8 Task the monkey was required to acquire targets alternating between a centre target and one of eight peripheral targets equidistantly spaced on the circumference of a circle. For our closed-loop BMI experiments, the peripheral targets were positioned 8 cm from the centre target. In hand-reaching data sets used for decoder training and offline decode, the targets were either 8 or 12 cm (the majority of data sets) from the centre. In much of Monkey L's training data, the three targets forming the upper quadrant were placed slightly further (13 and 14 cm) based on previous experience that this led to decoders with improved ability to acquire targets in that quadrant. To acquire a target, the monkey had to hold the cursor within a  $4 \text{ cm} \times 4 \text{ cm}$  acceptance window centred on the target for 500 ms. If the target was acquired successfully, the monkey received a liquid reward. If the target was not acquired within 5 s (BMI control) or 2 s (hand control) of target presentation, the trial was a failure and no reward was given.

Although the data included in this study span many months of each animal's research career, these data start after each animal was well-trained in performing point-to-point planar reaches; day-to-day variability when making the same reaching movements was modest. To quantify behavioural similarity across the study, we took advantage of having collected the same 'Baseline Block' task data at the start of most experimental sessions: 171/185 monkey R days, 398/452 monkey L days. This consisted of  $\sim 200$  trials of arm-controlled Radial 8 Task reaches, with targets 8 cm from the centre. For each of these recording sessions, we calculated the mean hand  $x$  and  $y$  velocities (averaged over trials to/from a given radial target) throughout a 700 ms epoch following radial target onset for outward reaches and 600 ms following centre target onset for inward reaches (inward reaches were slightly faster). We concatenated these velocity time series across the 8 different targets, producing 10,400 ms  $x$  velocity and  $y$  velocity vectors from each recording session. Behavioural similarity between any two recording sessions was then measured by the Pearson correlation between the data sets' respective  $x$  and  $y$  velocity vectors. Then, the two dimensions' correlations were averaged to produce a single-correlation value between each pair of sessions. These hand velocity correlations were  $0.90 \pm 0.04$  (mean  $\pm$  s.d. across days) for monkey R, and  $0.91 \pm 0.04$  for monkey L.

We measured closed-loop BMI performance on the Radial 8 Task using two metrics. Target acquisition rate is the number of peripheral targets acquired divided by the duration of the task. This metric holistically reflects cursor-control ability because, unlike time to target, it is negatively affected by failed trials and directly relates to the animal's rate of liquid reward. Targets per minute is calculated over all trials of an experimental condition (that is, which decoder was used) and therefore yields a single measurement per day/experimental condition. Across-days distributions of a given decoder's targets per minute performance were consistent with a normal distribution (Kolmogorov-Smirnov test), justifying our use of paired  $t$ -tests statistics when comparing this metric. This is consistent with the measure reflecting the accumulated outcome of many hundreds of random processes (individual trials). As a second measure of performance that is more sensitive when success rates are high and similar between decoders (such as the 'ideal' conditions where we presented no challenges to the decoders), we compared times to target. This measure consists of the time between when the target appeared and when the cursor entered the target acceptance window before successfully acquiring the target, but does not include the 500 ms hold time

(which is constant across all trials). Times to target are only measured for successful trials to peripheral targets, and were only compared when success rates were not significantly different (otherwise, a poor decoder with a low success rate that occasionally acquired a target quickly by chance could nonsensically ‘outperform’ a good decoder with 100% success rate but slower times to target). Because these distributions were not normal, we used the Mann–Whitney–Wilcoxon rank-sum tests when comparing two decoders’ times with target.

In the Random Target Task each trial’s target appeared at a random location within a 20 cm × 20 cm region centred within a larger workspace that was 40 × 30 cm. A new random target appeared after each trial regardless of whether this trial was a success or a failure due to exceeding the 5 s time limit. The target location randomization enforced a rule that the new target’s acceptance area could not overlap with that of the previous target. Performance on the Random Target Task was measured by success rate (the number of successfully acquired targets divided by the total number of presented targets) and the normalized time to target. Normalized time to target is calculated for successful trials following another successful trial, and is the duration between target presentation and target acquisition (not including the 500 ms hold time), divided by the straight-line distance between this target’s centre and the previously acquired target’s centre<sup>62</sup>.

**Decoder comparison experiment design.** All offline decoding comparisons between MRNN and FIT-KF were performed using test data that were held out from the data used to train the decoders. Thus, although the MRNN has many more parameters than FIT-KF, both of these fundamentally different algorithm types were trained according to best practices with matched training and test data. This allows their performance to be fairly compared. Decode accuracy was measured as the square of the Pearson’s correlation coefficient between true and decoded hand endpoint velocity in the fronto-parallel plane.

When comparing online decoder performance using BMI-controlled Radial 8 Target or Random Target Tasks, the decoders were tested using an interleaved block-set design in which contiguous ~200 trial blocks of each decoder were run followed by blocks of the next decoder, until the block-set comprising all tested decoders was complete and the next block-set began. For example, in the electrode-dropping experiments (Fig. 3), this meant an ‘AB AB’ design where A could be a block of MRNN trials and B could be a block of FIT Sameday trials. For the stale training data experiments (Fig. 4), an ‘ABCD ABCD ABCD...’ design was used to test the four different decoders. When switching decoders, we gave the monkey ~20 trials to transition to the new decoder before starting ‘counting’ performance in the block; we found this to be more than sufficient for both animals to adjust. For electrode-dropping experiments, the order of decoders within each block-set was randomized across days. For stale training data experiments, where several decoders often performed very poorly, we manually adjusted the order of decoders within block-sets so as to keep the monkeys motivated by alternating what appeared to be more and less frustrating decoders. All completed blocks were included in the analysis. Throughout the study, the experimenters knew which decoder was in use, but all comparisons were quantitative and performed by the same automated computer program using all trials from completed blocks. The monkeys were not given an overt cue to the decoder being used.

During online experiments, we observed that when a decoder performed extremely poorly, such that the monkey could not reliably acquire targets within the 5 s time limit, the animal stopped performing the task before the end of the decoder evaluation block. To avoid frustrating the monkeys, we stopped a block if the success rate fell below 50% after at least 10 trials. This criterion was chosen based on pilot studies in which we found that below this success rate, the monkey would soon thereafter stop performing the task and would frequently refuse to re-engage for a prolonged period of time. Our interleaved block design meant that each decoder was tested multiple times on a given experimental session, which in principle provides the monkey multiple attempts to finish a block with each decoder. In practice, we found that monkeys could either complete every block or no blocks with a given decoder, and we refer to decoders that could not be used to complete a block as having failed. The performance of these decoders was recorded as 0 targets per minute for that experimental session. The exception to the above was that during an electrode-dropping experiment session, we declared both FIT-KF Sameday and MRNN as having failed for a certain number of electrodes dropped if the monkey could not complete a block with either decoder. That is, we did not continue with a second test of both (unusable) decoders as per the interleaved block design, because this would have unduly frustrated the animal.

We performed this study with two monkeys, which is the conventional standard for systems neuroscience and BMI experiments using a non-human primate model. No monkeys were excluded from the study. We determined how many experimental sessions to perform as follows. For all offline analyses, we examined the dates of previous experimental sessions with suitable arm reaching data and selected sets of sessions with spacing most appropriate for each analysis (for example, closely spaced sessions for Fig. 2b, all of the available data for Fig. 2c, two clusters with a gap for stale training analyses). All these predetermined sessions were then included in the analysis. For the stale training data experiments (Fig. 4), the choice of two gaps with three test days each was pre-established. For the electrode-dropping experiments (Fig. 3), we did not know *a priori* how electrode dropping would affect performance and when each decoder would fail. We therefore determined the maximum number of electrodes to drop during the experiment and adjusted the number of sessions testing each drop condition during

the course of experiments to comprehensively explore the ‘dynamic range’ across which decoder robustness appeared to differ. For both of these experiments, during an experimental session additional block-sets were run until the animal became satiated and disengaged from the task. We did not use formal effect size calculations to make data sample size decisions, but did perform a variety of experiments with large numbers of decoder comparison trials (many tens of thousands) so as to be able to detect substantial decoder performance differences. For secondary online experiments (Supplementary Figs 3 and 7), which served to support offline analyses (Supplementary Fig. 3) or demonstrate that the MRNN could acquire other target locations (Supplementary Fig. 7), we chose to perform only 1–3 sessions per animal in the interest of conserving experimental time.

**Neural decoding using an MRNN.** At a high level, the MRNN decoder transforms inputs  $\mathbf{u}(t)$ , the observed spike counts on each electrode at a particular time, into a cursor position and velocity output. This is accomplished by first training the artificial recurrent neural network; that is, adjusting the weights of an artificial recurrent neural network such that when the network is provided a time series of neural data inputs, the data kinematic outputs can be accurately ‘read out’ from this neural network’s state. The rest of this section will describe the architecture, training and use of the MRNN for the purpose of driving a BMI.

The generic recurrent network model is defined by an  $N$ -dimensional vector of activation variables,  $\mathbf{x}$ , and a vector of corresponding ‘firing rates’,  $\mathbf{r} = \tanh \mathbf{x}$ . Both  $\mathbf{x}$  and  $\mathbf{r}$  are continuous in time and take continuous values. In the standard RNN model, the input affects the dynamics as an additive time-dependent bias in each dimension. In the MRNN model, the input instead directly parameterizes the artificial neural network’s recurrent weight matrix, allowing for a multiplicative interaction between the input and the hidden state. One view of this multiplicative interaction is that the hidden state of the recurrent network is selecting an appropriate decoder for the statistics of the current data set. The equation governing the dynamics of the activation vector is of the form suggested in ref. 35, but adapted in this study to continuous time to control the smoothness to MRNN outputs,

$$\tau \dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \mathbf{J}^{\mathbf{u}(t)} \mathbf{r}(t) + \mathbf{b}^{\mathbf{x}}.$$

The  $N \times N \times |\mathbf{u}|$  tensor  $\mathbf{J}^{\mathbf{u}(t)}$  describes the weights of the recurrent connections of the network, which are dependent on the  $E$ -dimensional input,  $\mathbf{u}(t)$ . The symbol  $|\mathbf{u}|$  denotes the number of unique values  $\mathbf{u}(t)$  can take. Such a tensor is unusable for continuous valued  $\mathbf{u}(t)$  or even discrete valued  $\mathbf{u}(t)$  with prohibitively many values. To make these computations tractable, the input is linearly combined into  $F$  factors and  $\mathbf{J}^{\mathbf{u}(t)}$  is factorized<sup>35</sup> according to the following formula:

$$\mathbf{J}^{\mathbf{u}(t)} = \mathbf{J}^{\mathbf{x}^f} \cdot \text{diag}(\mathbf{J}^{\mathbf{u}} \mathbf{u}(t)) \cdot \mathbf{J}^{\mathbf{x}},$$

where  $\mathbf{J}^{\mathbf{x}^f}$  has dimension  $N \times F$ ,  $\mathbf{J}^{\mathbf{u}}$  has dimension  $F \times E$ ,  $\mathbf{J}^{\mathbf{x}}$  has dimension  $F \times N$ , and  $\text{diag}(\mathbf{v})$  takes a vector,  $\mathbf{v}$ , and returns a diagonal matrix with  $\mathbf{v}$  along the diagonal. One can directly control the complexity of interactions by choosing  $F$ . In addition, the network units receive a bias  $\mathbf{b}^{\mathbf{x}}$ . The constant  $\tau$  sets the time scale of the network, so we set  $\tau$  in the physiologically relevant range of hundreds of milliseconds. The output of the network is read out from a weighted sum of the network firing rates plus a bias, defined by the equation

$$\mathbf{z}(t) = \mathbf{W}_o \mathbf{r}(t) + \mathbf{b}^{\mathbf{z}},$$

where  $\mathbf{W}_o$  is an  $M \times N$  matrix, and  $\mathbf{b}^{\mathbf{z}}$  is an  $M$ -dimensional bias.

**Table 1 | Network and training parameters used for the closed-loop MRNN BMI decoder.**

	Monkey R	Monkey L
$\Delta t$	20 ms	20–30 ms
$\tau$	100 ms	100–150 ms
$N$	100	50
$F$	100	50
$\sigma_{\text{trial}}$	0.045	0.045
$\sigma_{\text{electrode}}$	0.3	0.3
$g_{\text{xf}}$	1.0	1.0
$g_{\text{fu}}$	1.0	1.0
$g_{\text{fx}}$	1.0	1.0
$E$	192	96
Days of training data	82–129	189–230
Years spanned	1.59	2.77
Number of params in each MRNN	39502	9952
$\beta$	0.99	0.99

BMI, brain-machine interface; MRNN, multiplicative recurrent neural network.

**MRNN training.** We began decoder training by instantiating MRNNs of network size  $N = 100$  (monkey R) and  $N = 50$  (monkey L) with  $F = N$  in both cases (see Table 1 for all MRNN parameters). For monkey R, who was implanted with two multielectrode arrays,  $E = 192$ , while for monkey L with one array,  $E = 96$ . The non-zero elements of the non-sparse matrices  $\mathbf{J}^{xf}, \mathbf{J}^{fu}, \mathbf{J}^{fx}$  are drawn independently from a Gaussian distribution with zero mean and variance  $g_{xf}F, g_{fu}/E$ , and  $g_{fx}/N$ , with  $g_{xf}, g_{fu}$ , and  $g_{fx}$  set to 1.0 in this study. The elements of  $\mathbf{W}_o$  are initialized to zero, and the bias vectors  $\mathbf{b}^x$  and  $\mathbf{b}^z$  are also initialized to 0.

The input  $\mathbf{u}(t)$  to the MRNN (through the matrix  $\mathbf{J}^{u(t)}$ ) is the vector of binned spikes at each time step. Concatenating across time in a trial yields training data matrix,  $\mathbf{U}^j$ , of binned spikes of size  $E \times T^j$ , where  $T^j$  is the number of times steps for the  $j$ th trial. Data from five consecutive actual monkey-reaching trials are then concatenated together to make one 'MRNN training' trial. The first two actual trials in an MRNN training trial were used for seeding the hidden state of the MRNN (that is, not used for learning), whereas the next three actual trials were used for learning. With the exception of the first two actual trials from a given recording day, the entire set of actual trials are used for MRNN learning by incrementing the actual trial index that begins each training trial by one.

The parameters of the network were trained offline to reduce the averaged squared error between the measured kinematic training data and the output of the network,  $\mathbf{z}(t)$ . Specifically, we used the Hessian-Free (HF) optimization method<sup>36,63</sup> for RNNs (but adapted to the continuous-time MRNN architecture). HF is an exact second order method that uses back-propagation through time to compute the gradient of the error with respect to the network parameters. The set of trained parameters is  $\{\mathbf{J}^{xf}, \mathbf{J}^{fu}, \mathbf{J}^{fx}, \mathbf{b}^x, \mathbf{W}_o, \mathbf{b}^z\}$ . The HF algorithm has three critical parameters: the minibatch size; the initial lambda setting; and the max number of conjugate-gradient iterations. We set these parameters to one-fifth the total number of trials, 0.1 and 50, respectively. The optimizations were run for 200 steps and a snapshot of the network was saved every 10 steps. Among these snapshots, the network with the lowest cross-validation error on held-out data was used in the experiment.

We independently trained two separate MRNN networks to each output a 2D ( $M=2$ ) signal,  $\mathbf{z}(t)$ . The first network learned to output the normalized hand position through time in both the horizontal ( $x$ ) and vertical ( $y$ ) spatial dimensions. The second MRNN learned to output the hand velocity through time, also in the  $x$  and  $y$  dimensions. As training data for the velocity decoder, we calculated hand velocities from the hand positions numerically using central differences.

In this study, we trained a new MRNN whenever adding new training data; this allowed us to verify that the training optimization consistently converged to a high-quality decoder. However, it is easy to iteratively update an MRNN decoder with new data without training from scratch. By adding the new data to the training corpus and using the existing decoder weights as the training optimization's initial conditions, the MRNN will more rapidly converge to a new high-quality decoder.

**Training an MRNN with many data sets and perturbed inputs.** A critical element of achieving both high performance and robustness in the MRNN decoder was training the decoder using data from many previous recording days spanning many months. When training data sets included data from  $> 1$  day, we randomly selected a small number of trials from each day for a given minibatch. In this way, every minibatch of training data sampled the input distributions from all training days.

A second key element of training robustness to recording condition changes was a form of data augmentation in which we intentionally introduced perturbations to the neural spike trains that were used to train the MRNN. The concatenated input,  $\tilde{\mathbf{U}} = [\mathbf{U}^1, \dots, \mathbf{U}^{i+4}]$  was perturbed by adding and removing spikes from each electrode. We focus on electrode  $c$  of the  $j$ th training trial, that is, a row vector of data  $\tilde{\mathbf{U}}_{c,\dots}^j$ . Let the number of actual observed spikes in  $\tilde{\mathbf{U}}_{c,\dots}^j$  be  $n_c^j$ . This number was perturbed according to

$$\hat{n}_c^j = \eta^j \eta_c n_c^j,$$

where both  $\eta^j$  and  $\eta_c$  are Gaussian variables with a mean of one and s.d. of  $\sigma_{\text{trial}}$  and  $\sigma_{\text{electrode}}$ , respectively. Conceptually,  $\eta^j$  models a global firing rate modulation across all electrodes of the array (for example, array movement and arousal), while  $\eta_c$  models electrode by electrode perturbations such as electrode dropping or moving baselines in individual neurons. If  $\hat{n}_c^j$  was  $< 0$  or  $> 2n_c^j$ , it was resampled, which kept the average number of perturbed spikes in a given electrode and training trial roughly equal to the average number of true (unperturbed) spikes in the same electrode and training trial. Otherwise, if  $\hat{n}_c^j$  was greater than  $n_c^j$ , then  $\hat{n}_c^j - n_c^j$  spikes were added to random time bins of the training trial. If  $\hat{n}_c^j$  was less than  $n_c^j$ , then  $n_c^j - \hat{n}_c^j$  spikes were randomly removed from time bins of the training trial that already had spikes. Finally, if  $\hat{n}_c^j = n_c^j$ , nothing was changed.

The process of perturbing the binned spiking data occurred anew on every iteration of the optimization algorithm, that is, in the HF algorithm, the perturbation  $\hat{n}_c^j = \eta^j \eta_c n_c^j$  occurs after each update of the network parameters. Note that these input data perturbations were only applied during MRNN training; when the MRNN was used for closed-loop BMI control, true neural spike counts were provided as inputs. Supplementary Figure 3 shows the closed-loop control quality difference between the MRNN trained with and without this data augmentation. Our data augmentation procedure is reminiscent of dropout<sup>64</sup>, however our data perturbations are tailored to manage the nonstationarities in data associated with BMI.

**Controlling a BMI cursor with MRNN output.** Once trained, the MRNNs were compiled into the embedded real-time operating system and run in closed-loop to provide online BMI cursor control. The decoded velocity and position were initialized to 0, as was the MRNN hidden state. Thereafter, at each decode time step the parallel pair of MRNNs received binned spike counts as input and had their position and velocity outputs blended to yield a position estimate. This was used to update the drawn cursor position. The on-screen position that the cursor moves to during BMI control,  $d_x(t), d_y(t)$ , is defined by

$$d_x(t) = \beta(d_x(t - \Delta t) + \gamma_v v_x(t - \Delta t)\Delta t) + (1 - \beta)\gamma_p p_x(t)$$

$$d_y(t) = \beta(d_y(t - \Delta t) + \gamma_v v_y(t - \Delta t)\Delta t) + (1 - \beta)\gamma_p p_y(t)$$

where  $v_x, v_y, p_x, p_y$  are the normalized velocity and positions in the  $x$  and  $y$  dimensions and  $\gamma_v, \gamma_p$  are factors that convert from the normalized velocity and position, respectively, to the coordinates of the virtual-reality workspace. The parameter  $\beta$  sets the amount of position versus velocity decoding and was set to 0.99. In effect, the decode was almost entirely dominated by velocity, with a slight position contribution to stabilize the cursor in the workplace (that is, offset accumulated drift). Note that when calculating offline decode accuracy (Fig. 2), we set  $\beta$  to 1 to more fairly compare the MRNN to the FIT-KF decoder, which decodes velocity only.

We note that although (1) the MRNN's recurrent connections mean that previous inputs affect how subsequent near-term inputs are processed, and (2) our standard procedure was to retrain the MRNN with additional data after each experimental session, the MRNN is *not* an 'adaptive' decoder in the traditional meaning of the term. Its parameters are fixed during closed-loop use, and therefore when encountering recording condition changes, the MRNN cannot 'learn' from this new data to update its neural-to-kinematic mappings in the way that adaptive decoders do (for example, refs 4,24,27). Insofar as its architecture and training regime make the MRNN robust to input changes, this robustness is 'inherent' rather than 'adaptive.'

**Neural decoding using a FIT-KF.** We compared the performance of the MRNN with FIT-KF<sup>40</sup>. The FIT-KF is a Kalman filter where the underlying kinematic state,  $\mathbf{z}(t)$ , comprises the position and velocity of the cursor as well as a bias term. Observations of the neural binned spike counts,  $\mathbf{y}(t)$ , are used to update the kinematic state estimate. With  $\Delta t$  denoting bin width (25 ms in this study), the FIT-KF assumes the kinematic state gives rise to the neural observations according to the following linear dynamical system:

$$\mathbf{z}(t + \Delta t) = \mathbf{A}\mathbf{z}(t) + \mathbf{w}(t)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{z}(t) + \mathbf{q}(t)$$

where  $\mathbf{w}(t)$  and  $\mathbf{q}(t)$  are zero-mean Gaussian noise with covariance matrices  $\mathbf{W}$  and  $\mathbf{Q}$ , respectively. The Kalman filter is a recursive algorithm that estimates the state  $\mathbf{z}(t)$  using the current observation  $\mathbf{y}(t)$  and the previous state estimate  $\mathbf{z}(t - \Delta t)$ . Previous studies have used such decoders to drive neural cursors (for example refs 5,38,65).

The parameters of this linear dynamical system,  $\mathbf{A}, \mathbf{W}, \mathbf{C}, \mathbf{Q}$ , are learned in a supervised manner from hand reach training data using maximum-likelihood estimation, further described in refs 6,66. The FIT-KF then incorporates two additional innovations. First, it performs a rotation of the training kinematics using the assumption that at every moment in time, the monkey intends to move the cursor directly towards the target. Second, it assumes that at every time step, the monkey has perfect knowledge of the decoded position via visual feedback. This affects Kalman filter inference in two ways: first, the covariance of the position estimate in Kalman filtering is set to 0; and second, the neural activity that is explainable by the cursor position is subtracted from the observed binned spike counts. These innovations are further described in refs 6,40.

**Mutual information for determining electrode-dropping order.** When testing the decoders' robustness to unexpected electrode loss, we determined which electrodes to drop by calculating the mutual information between each electrode's binned spike counts and the reach direction. This metric produced a ranking of electrodes in terms of how statistically informative they were of the reach direction; importantly, this metric is independent of the decoder being used. Let  $p$  denote the distribution of an electrode's binned firing rates,  $y$  denote the binned spike counts lying in a finite set  $Y$  of possible binned spike counts,  $M$  denote the number of reach directions and  $x_j$  denote reach direction  $j$ . The set  $Y$  comprised  $\{0, 1, 2, 3, 4, 5 + \}$  spike counts, where any spike counts greater than or equal to 5 were counted towards the same bin ('5 +'), corresponding to an instantaneous firing rate of 250 Hz in a 20 ms bin). We calculated the entropy of each electrode,

$$H(Y) = - \sum_{y \in Y} p(y) \log p(y),$$

as well as its entropy conditioned on the reach direction

$$H(Y|X) = - \sum_{j=1}^M p(x_j) \sum_{y \in Y} p(y|x_j) \log p(y|x_j).$$

From these quantities, we calculated the mutual information between the neural

activity and the reach direction as  $I_{\text{drop}}(X;Y) = H(Y) - H(Y|X)$ . We dropped electrodes in order from highest to lowest mutual information.

**Principal angles of neural subspaces analysis.** For a parsimonious scalar metric of how similar patterns of neural activity during reaching were between a given pair of recording days (used in Supplementary Fig. 1), we calculated the minimum principal angle between the neural subspaces of each recording day. We defined the neural subspace on a recording day as the top  $K$  principal components of the neural coactivations. Put more simply, we asked how similar day  $i$  and day  $j$ 's motifs of covariance between electrodes' activity were during arm reaching. Specifically, we started with a matrix  $Y_i$  from each day  $i$  consisting of neural activity collected while the monkey performed  $\sim 200$  trials of a Radial 8 Task (8 cm distance to targets) using arm control; this task has been run at the start of almost every experimental session conducted using both monkeys R and L since array implantation.  $Y_i$  is of dimensionality  $E \times T$ , where  $E$  is the number of electrodes and  $T$  is the number of non-overlapping 20 ms bins comprising the duration of this task. We next subtracted from each row of  $Y_i$  that electrode's across-days mean firing rate (we also repeated this analysis without across-days mean subtraction and observed qualitatively similar results, not shown). To obtain the principal components, we performed eigenvalue decomposition on the covariance matrix  $Y_i Y_i^T$  (note,  $Y_i$  is zero mean), and defined the matrix  $V_i$  as the first  $K$  eigenvectors.  $V_i$  had dimensions  $E \times K$ , where each column  $k$  is the vector of principal component coefficients (eigenvector) corresponding to the  $k$ th largest eigenvalue of the decomposition. Supplementary Figure 1 was generated using  $K = 10$ , that is, keeping the first 10 PCs, but the qualitative appearance of the data were similar when  $K$  was varied from 2 to 30 (not shown). Finally, the difference metric between days  $i$  and  $j$  was computed as the minimum of the  $K$  subspace angles between matrices  $V_i$  and  $V_j$ . Subspace angles were computed using the *subspacea* MATLAB function<sup>67</sup>.

**Data availability.** All relevant data and analysis code can be made available by the authors on request.

## References

- Hochberg, L. R. *et al.* Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature* **485**, 372–375 (2012).
- Collinger *et al.* High-performance neuroprosthetic control by an individual with tetraplegia. *Lancet* **381**, 557–564 (2013).
- Gilja, V. *et al.* Clinical translation of a high-performance neural prosthesis. *Nat. Med.* **21**, 1142–1145 (2015).
- Jarosiewicz, B. *et al.* Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Sci. Transl. Med.* **7**, 313ra179–313ra179 (2015).
- Bacher, D. *et al.* Neural point-and-click communication by a person with incomplete locked-in syndrome. *Neurorehabil. Neural Repair* **29**, 462–471 (2015).
- Gilja, V. *et al.* A high-performance neural prosthesis enabled by control algorithm design. *Nat. Neurosci.* **15**, 1752–1757 (2012).
- Nuyujukian, P., Fan, J. M., Kao, J. C., Ryu, S. I. & Shenoy, K. V. A high-performance keyboard neural prosthesis enabled by task optimization. *IEEE Trans. Biomed. Eng.* **62**, 21–29 (2015).
- Ganguly, K. & Carmena, J. M. Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biol.* **7**, e1000153 (2009).
- Flint, R. D., Ethier, C., Oby, E. R., Miller, L. E. & Slutzky, M. W. Local field potentials allow accurate decoding of muscle activity. *J. Neurophysiol.* **108**, 18–24 (2012).
- Nuyujukian, P. *et al.* Performance sustaining intracortical neural prostheses. *J. Neural Eng.* **11**, 66003 (2014).
- Chestek, C. A. *et al.* Single-neuron stability during repeated reaching in macaque premotor cortex. *J. Neurosci.* **27**, 10742–10750 (2007).
- Simeral, J. D., Kim, S.-P., Black, M. J., Donoghue, J. P. & Hochberg, L. R. Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array. *J. Neural Eng.* **8**, 25027 (2011).
- Perge, J. A. *et al.* Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. *J. Neural Eng.* **10**, 36004 (2013).
- Perge, J. A. *et al.* Reliability of directional information in unsorted spikes and local field potentials recorded in human motor cortex. *J. Neural Eng.* **11**, 46007 (2014).
- Wodlinger, B. *et al.* Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations. *J. Neural Eng.* **12**, 16011 (2015).
- Aflalo, T. *et al.* Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science* **348**, 906–910 (2015).
- Jarosiewicz, B. *et al.* Advantages of closed-loop calibration in intracortical brain-computer interfaces for people with tetraplegia. *J. Neural Eng.* **10**, 46012 (2013).
- Huggins, J. E., Wren, P. A. & Gruis, K. L. What would brain-computer interface users want? Opinions and priorities of potential users with amyotrophic lateral sclerosis. *Amyotroph. Lateral Scler.* **12**, 318–324 (2011).
- Collinger *et al.* Functional priorities, assistive technology, and brain-computer interfaces after spinal cord injury. *J. Rehabil. Res. Dev.* **50**, 145 (2013).
- Srinivasan, L., Eden, U. T., Mitter, S. K. & Brown, E. N. General-purpose filter design for neural prosthetic devices. *J. Neurophysiol.* **98**, 2456–2475 (2007).
- Wu, W. & Hatsopoulos, N. G. Real-time decoding of nonstationary neural activity in motor cortex. *IEEE Trans. Neural Syst. Rehabil. Eng.* **16**, 213–222 (2008).
- Shpigelman, L., Lalazar, H. & Vaadia, E. Kernel-ARMA for hand tracking and brain-machine interfacing during 3D motor control. *Neural Inf. Process. Syst.* 1489–1496 (2008).
- DiGiovanna, J., Mahmoudi, B., Fortes, J., Principe, J. C. & Sanchez, J. C. Coadaptive brain-machine interface via reinforcement learning. *IEEE Trans. Biomed. Eng.* **56**, 54–64 (2009).
- Li, Z., O'Doherty, J. E., Lebedev, M. A. & Nicolelis, M. A. L. Adaptive decoding for brain-machine interfaces through bayesian parameter updates. *Neural Comput.* **23**, 3162–3204 (2011).
- Dangi, S., Gowda, S., Heliot, R. & Carmena, J. M. in *2011 5th International IEEE/EMBS Conference on Neural Engineering (NER)* 609–612 (Cancun, Mexico, 2011).
- Mahmoudi, B. & Sanchez, J. C. A symbiotic brain-machine interface through value-based decision making. *PLoS ONE* **6**, e14760 (2011).
- Orsborn, A. L. *et al.* Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron* **82**, 1380–1393 (2014).
- Bishop, W. *et al.* Self-recalibrating classifiers for intracortical brain-computer interfaces. *J. Neural Eng.* **11**, 26001 (2014).
- Pohlmeyer, E. A., Mahmoudi, B., Geng, S., Prins, N. W. & Sanchez, J. C. Using reinforcement learning to provide stable brain-machine interface control despite neural input reorganization. *PLoS ONE* **9**, e87253 (2014).
- Chestek, C. A. *et al.* Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. *J. Neural Eng.* **8**, 45005 (2011).
- Flint, R. D., Wright, Z. A., Scheid, M. R. & Slutzky, M. W. Long term, stable brain machine interface performance using local field potentials and multiunit spikes. *J. Neural Eng.* **10**, 56005 (2013).
- Stavisky, S. D., Kao, J. C., Nuyujukian, P., Ryu, S. I. & Shenoy, K. V. A high performing brain-machine interface driven by low-frequency local field potentials alone and together with spikes. *J. Neural Eng.* **12**, 36009 (2015).
- Homer, M. L., Nurmikko, A. V., Donoghue, J. P. & Hochberg, L. R. Sensors and decoding for intracortical brain computer interfaces. *Annu. Rev. Biomed. Eng.* **15**, 383–405 (2013).
- Kao, J. C., Stavisky, S. D., Sussillo, D., Nuyujukian, P. & Shenoy, K. V. Information systems opportunities in brain-machine interface decoders. *Proc. IEEE* **102**, 666–682 (2014).
- Sutskever, I., Martens, J. & Hinton, G. in *Proceedings of the 28th International Conference on Machine Learning (ICML)* (Bellevue, WA, USA, 2011).
- Martens, J. & Sutskever, I. in *Proceedings of the 28th International Conference on Machine Learning (ICML)* 1033–1040 (Bellevue, WA, USA, 2011).
- Churchland, M. M. & Shenoy, K. V. Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *J. Neurophysiol.* **97**, 4235–4257 (2007).
- Sussillo, D. *et al.* A recurrent neural network for closed-loop intracortical brain-machine interface decoders. *J. Neural Eng.* **9**, 26027 (2012).
- Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **9**, 1735–1780 (1997).
- Fan, J. M. *et al.* Intention estimation in brain-machine interfaces. *J. Neural Eng.* **11**, 16004 (2014).
- Chase, S. M., Schwartz, A. B. & Kass, R. E. Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. *Neural Netw.* **22**, 1203–1213 (2009).
- Koyama, S. *et al.* Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control. *J. Comput. Neurosci.* **29**, 73–87 (2010).
- Cunningham, J. P. *et al.* A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. *J. Neurophysiol.* **105**, 1932–1949 (2011).
- Seo, D. *et al.* Wireless recording in the peripheral nervous system with ultrasonic neural dust neuron neuroresource wireless recording in the peripheral nervous system with ultrasonic neural dust. *Neuron* **91**, 1–11 (2016).
- Fraser, G. W., Chase, S. M., Whitford, A. & Schwartz, A. B. Control of a brain-computer interface without spike sorting. *J. Neural Eng.* **6**, 55004 (2009).
- So, K., Dangi, S., Orsborn, A. L., Gastpar, M. C. & Carmena, J. M. Subject-specific modulation of local field potential spectral power during brain-machine interface control in primates. *J. Neural Eng.* **11**, 26002 (2014).
- Shenoy, K. V. & Carmena, J. M. Combining decoder design and neural adaptation in brain-machine interfaces. *Neuron* **84**, 665–680 (2014).

48. Barrese, J. C. *et al.* Failure mode analysis of silicon-based intracortical microelectrode arrays in non-human primates. *J. Neural Eng.* **10**, 66014 (2013).
49. Gold, C., Henze, D. A., Koch, C. & Buzsáki, G. On the origin of the extracellular action potential waveform: A modeling study. *J. Neurophysiol.* **95**, 3113–3128 (2006).
50. Kao, J. C. *et al.* in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) 2013*, 293–298 (IEEE, 2013).
51. Cunningham, J. P. *et al.* A closed-loop human simulator for investigating the role of feedback-control in brain-machine interfaces. *J. Neurophysiol.* **105**, 1932–1949 (2010).
52. Schwartz, A., Ketter, R. & Georgopoulos, A. Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement. *J. Neurosci.* **8**, 2913–2927 (1988).
53. Riehle, A. & Requin, J. Monkey primary motor and premotor cortex: single-cell activity related to prior information about direction and extent of an intended movement. *J. Neurophysiol.* **61**, 534–549 (1989).
54. Wise, S. P., Moody, S. L., Blomstrom, K. J. & Mitz, A. R. Changes in motor cortical activity during visuomotor adaptation. *Exp. Brain Res.* **121**, 285–299 (1998).
55. Crammond, D. & Kalaska, J. Differential relation of discharge in primary motor cortex and premotor cortex to movements versus actively maintained postures during a reaching task. *Exp. Brain Res.* **108**, 45–61 (1996).
56. Messier, J. & Kalaska, J. F. Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task. *J. Neurophysiol.* **84**, 152–165 (2000).
57. Stark, E. & Abeles, M. Predicting movement from multiunit activity. *J. Neurosci.* **27**, 8387–8394 (2007).
58. Perel, S. *et al.* Single-unit activity, threshold crossings, and local field potentials in motor cortex differentially encode reach kinematics. *J. Neurophysiol.* **114**, 1500–1512 (2015).
59. Christie, B. P. *et al.* Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance. *J. Neural Eng.* **12**, 16009 (2015).
60. Nuyujukian, P. *et al.* in *33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1301–1305 (Boston, USA, 2011).
61. Kaufman, M. T., Churchland, M. M., Ryu, S. I. & Shenoy, K. V. Cortical activity in the null space: permitting preparation without movement. *Nat. Neurosci.* **17**, 440–448 (2014).
62. Suminski, A. J., Tkach, D. C., Fagg, A. H. & Hatsopoulos, N. G. Incorporating feedback from multiple sensory modalities enhances brain-machine interface control. *J. Neurosci.* **30**, 16777–16787 (2010).
63. Martens, J. in *27th International Conference on Machine Learning* **951**, 735–742 (Haifa, Israel, 2010).
64. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. Preprint at arXiv:1207.0580. (2012).
65. Kim, S.-P., Simeral, J. D., Hochberg, L. R., Donoghue, J. P. & Black, M. J. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *J. Neural Eng.* **5**, 455–476 (2008).
66. Wu, W. *et al.* Neural decoding of cursor motion using a Kalman filter. *Adv. Neural Inf. Process. Syst.* **15**, 133–140 (2003).
67. Knyazev, A. V. & Argentati, M. E. Principal angles between subspaces in an a-based scalar product: algorithms and perturbation estimates. *SIAM J. Sci. Comput.* **23**, 2008–2040 (2002).

## Acknowledgements

We thank Mackenzie Risch, Michelle Wechsler, Liana Yates, Shannon Smith and Rosie Steinbach for surgical assistance and veterinary care; Evelyn Castaneda and Beverly Davis for administrative support; Boris Oskotsky for information technology support.

## Author contributions

D.S., S.D.S. and J.C.K. designed the experiments. D.S. trained the MRNN decoders. S.D.S. and J.C.K. conducted the experiments and data analysis. S.D.S. wrote the manuscript. J.C.K., D.S. and S.I.R. assisted in manuscript preparation. S.I.R. was responsible for surgical implantation. K.V.S. was involved in all aspects of experimentation, data review and manuscript preparation.

## Additional information

**Supplementary Information** accompanies this paper at <http://www.nature.com/naturecommunications>

**Competing financial interests:** The authors declare no competing financial interests.

**Reprints and permission** information is available online at <http://npg.nature.com/reprintsandpermissions/>

**How to cite this article:** Sussillo, D. *et al.* Making brain–machine interfaces robust to future neural variability. *Nat. Commun.* **7**, 13749 doi: 10.1038/ncomms13749 (2016).

**Publisher's note:** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



This work is licensed under a Creative Commons Attribution 4.0 International License. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in the credit line; if the material is not included under the Creative Commons license, users will need to obtain permission from the license holder to reproduce the material. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>

© The Author(s) 2016