Neural Embeddings Rank: Aligning 3D latent dynamics with movements

Chenggang Chen, Zhiyu Yang, Xiaoqin Wang
Department of Biomedical Engineering, Johns Hopkins University
`cheng-gang.chen@jhu.edu`

Abstract

Aligning neural dynamics with movements is a fundamental goal of neuroscience and brain-machine interfaces. However, we still lack a dimensionality reduction method that can align low-dimensional latent dynamics with movement. To fill this gap, we propose Neural Embedding Ranks (NER), which embed neural dynamics into a 3D latent space and contrast the embeddings based on movement ranks. Essentially, NER learns to *regress* **continuous** representations of neural dynamics (i.e., embeddings) on **continuous** movement. We apply NER and six other dimensionality reduction techniques to neurons in the primary motor cortex (M1), dorsal premotor cortex (PMd), and primary somatosensory cortex (S1) as monkeys perform reaching tasks. Only NER aligns latent dynamics with both hand position and direction, visualizable in 3D. NER reveals consistent latent dynamics in M1 and PMd across sixteen sessions over one year. A linear regression decoder with NER explains 86% and 97% of the variance in velocity and position, respectively. Linear models trained on data from one session can decode velocity, position, and direction in held-out test data from different dates and areas (64%, 88%, and 90%). NER also reveals distinct latent dynamics in S1 during consistent movements and in M1 when the monkey performs curved reaching tasks. The code is available at https://github.com/NeuroscienceAI/NER.

1 Introduction

It has long been thought that individual neurons in the motor and premotor cortex, similar to those in the visual and somatosensory cortex, are tuned to specific movement parameters such as direction. However, this static and receptive field-based neural representation fails to explain movement trajectories during simple tasks like reaching. Recent studies have found that the activities of multiple simultaneously recorded neurons, which fire spikes in a time-dependent manner, encode reaching movements [7]. Unlike the one-dimensional dynamics from a single neuron, understanding how movements are represented by these high-dimensional neural dynamics is challenging. In systems neuroscience and brain-machine interfaces, there is significant motivation to reduce these high-dimensional neural dynamics to lower-dimensional latent dynamics. First, visualizing invisible high-dimensional neural dynamics. This involves a trade-off between dimensionality and explained variance. To explain a complex reaching task, at least six dimensions are typically required. For example, in an eight-direction center-out reaching task, [8][6][9] selects fifteen dimensions for the PMd, ten for the M1, and eight for the S1. Therefore, we need to further reduce the dimensionality of these "low-dimensional" latent dynamics. Currently, we still lack a method that can directly explain enough variance within three dimensions. Second, comparing movements with latent dynamics. After dimensionality reduction, we can visualize the trajectories of latent dynamics over time. For instance, [5] reveals rotational latent dynamics during reaching tasks using principal component analysis (PCA). [20] found that animals performing similar tasks exhibit similar latent dynamics. However, these trajectories do not align precisely with the reaching movements: when the hand

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

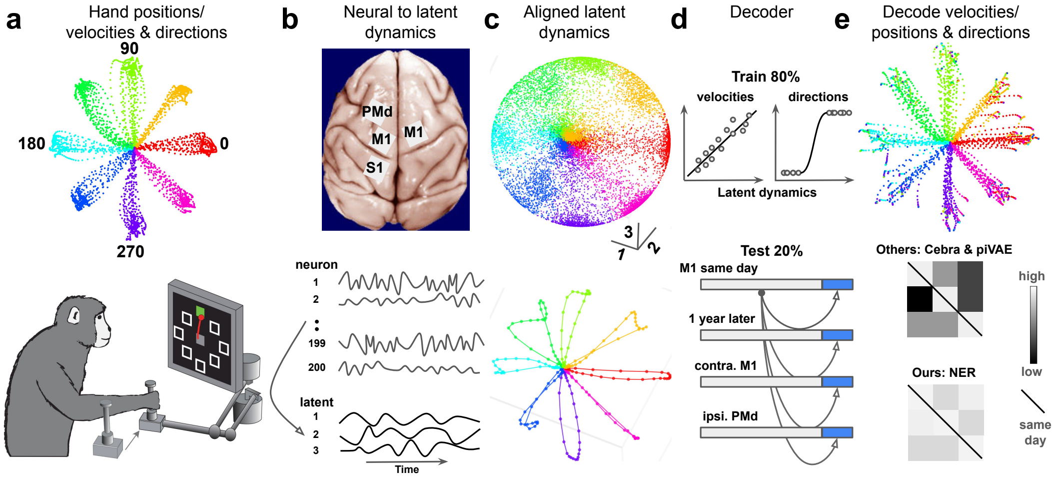

Figure 1: NER aligns 3D latent dynamics with movements and enabling cross sessions movement decoding. **a** Top: The monkey performs a center-out reaching task in eight directions using a planar manipulandum. Hand velocity is computed from hand position. Bottom: The monkey moves the cursor to outer targets to receive rewards. **b** Top: Neural dynamics are recorded using a 96-channel Utah array in two monkeys. Monkey C has implants in the M1 of the right hemisphere first and in M1 and PMd of the left hemisphere second. Monkey L has an implant in area 2 of the S1 of the left hemisphere. During task, all monkeys use the hand contralateral to the implanted hemisphere. Bottom: Spiking activities from multiple neurons (44 to 211) are recorded during multiple trials (168 to 1038) of the behavioral task. Dimensionality reduction reduces 200-dimensional neural dynamics to three-dimensional latent dynamics. **c** Top: Neural dynamics from 190 dimensions (neurons) in the PMd are reduced to 3D latent dynamics. The figure includes 7600 dots from 190 trials with 40 bins each. Bottom: Trial-averaged latent dynamics. Data is from Monkey C (date: 2016-10-14). **d** Top: Linear and logistic regression decoders are trained on the same independent variables (latent dynamics) but different dependent variables (hand velocities and directions) using 80% of trial data. Bottom: The trained decoder from 80% of trial data is used to predict movements on the 20% held-out test data. The model trained on one day predicts movements across one year, in the contralateral M1, and ipsilateral PMd. **e** Top: Two linear decoders trained from **c** decode hand velocities (positions) and directions with R-squared accuracy of 86% (96%) and peak accuracy of 97%, respectively, on the 20% held-out test data. Bottom: Decoders trained on the same day (diagonal) have much better decoding performance than models trained on different conditions (off-diagonal) using previous dimensionality reduction methods, whereas our method has higher and consistent decoding performance.

reaches in eight directions, the trajectories of latent dynamics are neither in eight distinct directions nor well-separated, often appearing entangled. Third, decoding movement using latent dynamics. Decoders trained on individual neural activities are commonly used to predict movements [10]. However, a drawback of using individual neural activities is that when the identities of neurons change during long-term recordings, the decoding performance deteriorates [6]. Additionally, it is challenging to decode movement from different brain areas or across different animals. Decoders trained on latent dynamics facilitate long-term or cross-animal decoding. Since latent dynamics do not fully capture neural dynamics, decoding performance is often suboptimal with linear decoders, necessitating the use of nonlinear decoders or deep neural networks. Thus, decoding movement using a linear decoder without hyperparameters remains a significant challenge.

As our goal is to extract the latent dynamics most informative about movements, we have decided to train the latent dynamics using movements as the target. Several recent studies have already trained latent dynamics to classify different movement directions or positions using variational auto-encoders (VAE) [26][11][15] or contrastive learning[22][1]. In this paper, we are inspired by the fact that many features, including movements, are continuous, and a major task of many neurons is not classification but regression. For example, neurons display monotonic tunings to light intensity and sound levels [3][19]. Even for discrete features like faces and objects, neurons in inferotemporal cortex encode continuous feature dimensions and can decode face identities using linear regression [2][4]. Therefore, we trained latent dynamics to regress movement trajectories by minimizing the ranking loss. We

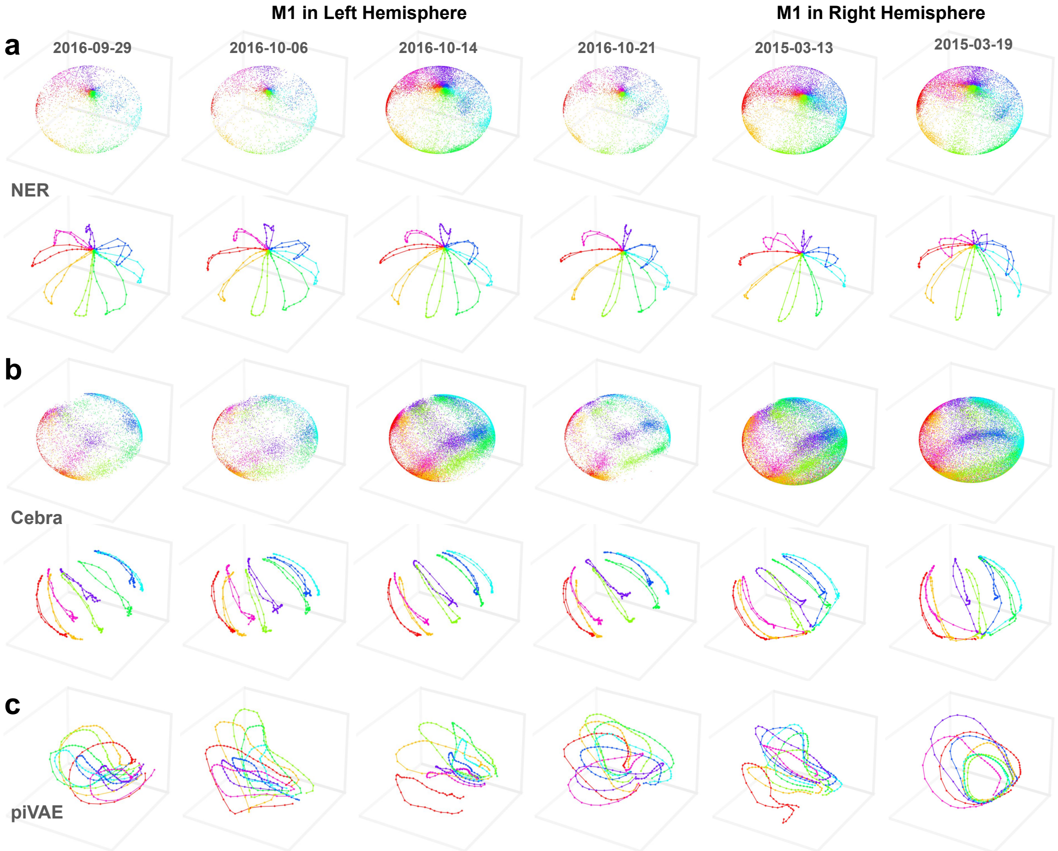

Figure 2: NER reveals consistent and movements aligned latent dynamics in M1. **a** Single trial (top) and averaged (bottom) latent dynamics from six sessions across one year in two hemispheres. Latent dynamics are rotated with reference to 2016-10-14 session using one of the eight reaching directions. **b-c**, similar to **a** but using Cebra and piVAE. Fig 8 shows remaining four sessions' latent dynamics of same monkey C. Fig 9 shows the single trial and/or averaged latent dynamics revealed by five other methods. Fig 10 show the trial-averaged latent dynamics before rotation. Fig 11 shows the entangled neural embeddings using PCA and time-stimulus components revealed by dPCA.

demonstrate that our method can reveal three-dimensional, behaviorally aligned latent dynamics. The revealed latent dynamics are highly informative and consistent in M1 and PMd but not in S1. Related work, motivation, our model, results in PMd, S1, curved movements in M1, network training, and datasets are given in Appendix. Fig 1 visualizes the pipeline used in this study.

2 Results

2.1 Movements aligned latent dynamics are consistent over years in M1

Across all ten sessions in the M1 of left and right hemisphere, NER consistently reveals neural embeddings that are aligned with movement (Fig 2a, Fig 8a). Importantly, on the initial movement stage, latent dynamics converge on the same starting points and form a pinwheel structure that resemble ground truth movements. Furthermore, we found almost the same neural embeddings in both hemispheres even if the data collection were separated over one year. Cebra is the 2nd best method that found comparable latent dynamics where both directions and positions are roughly aligned with movements, for both single and averaged embeddings (Fig 2b, Fig 8b). However, there are two limitations for Cebra: the movement starting points are widely separated which is different from ground truth movements, and less consistent latent dynamics cross sessions. For example, it revealed connected latent dynamics at the movements starting points on only two sessions (Fig 2b). piVAE is the 3rd best method where it has rough direction aligned latent dynamics at different directions and relatively separated single-trial neural embeddings (Fig 2c, Fig 8c). However, the latent dynamics is just correlated with movements but not aligned, and is less consistent across sessions. UMAP with labels shows clearly clustered neural embeddings corresponds different angles whereas UMAP without labels shows extended instead of higher clustered latent dynamics (Fig 9). Both models fail to generate aligned and consistent latent dynamics. PCA and dPCA also generate identifiable latent dynamics (Fig 9) and dPCA reveals both time and stimulus components (Fig 9). A major limitation of both methods is the mixed single trials neural embeddings (Fig 9).

Together, NER represent the best method for revealing movements aligned latent dynamics. We will further examine its ability in PMd and S1, and leverage the aligned latent dynamics for decoding movements within and across sessions, and exploring movement specific latent dynamics.

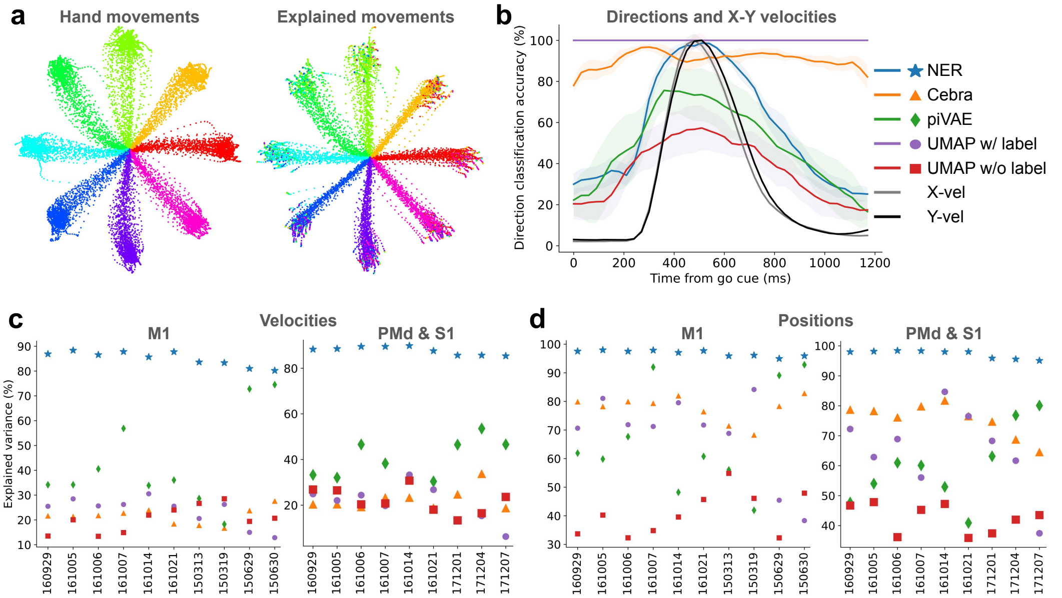

Figure 3: Explained variance of movements in M1, PMd, and S1. **a** Left, hand movement trajectories. Right, predicted trajectories by two decoders. The data is collected from PMd on 2016-10-14. The explained variance for the velocities and positions are 90% and 98%, respectively. **b** Hand direction classification accuracy using a logistic regression decoder. Shaded areas are standard deviation over six sessions from M1. **c** Linear regression decoder explained variance of hand velocities using the latent dynamics revealed by five dimensionality reduction methods (different color and shape). X-axis indicates the date of different sessions. Left, ten sessions from the M1 of Monkey C (six on the left and four on the right). Right, six sessions from PMd of Monkey C and four sessions from S1 of Monkey H. **d** Similar to **c** but for hand positions. Fig 12 shows the direction tuning curve in PMd, the correlation between tuning curves and velocities, and explained variance on directions.

2.2 Explained variance of movements using linear decoders in M1, PMd, and S1

Five dimensionality reduction methods reveal the single trial latent dynamics that are dependent on movements, next we will use those latent dynamics as the independent variables to explain the variance of dependent variables, i.e., hand directions, velocities and positions.

Fig 3a shows the ground truth and predicted hand movement trajectories using latent dynamics revealed by NER in PMd. A linear regression decoder can explain 90% and 98% variance in hand velocities and positions, respectively. In both M1 and PMd (Fig 3b, Fig 12a), a logistic regression decoder reveals tuning of direction classification accuracy from the start of go cue to the end of animal reach target. This tuning curve is highly correlated with hand velocities using the latency dynamics extracted by NER but not Cebra (M1: 0.93 vs 0.28, PMd: 0.93 vs 0.13. Fig 12b). NER outperforms four other methods in all sessions for explaining the variance of both hand velocities and positions (Fig 3c, d). For example, across ten sessions in the M1, the NER explains 86% of variance in velocities whereas the 2nd best model piVAE only explains 35% variance. We observed similar

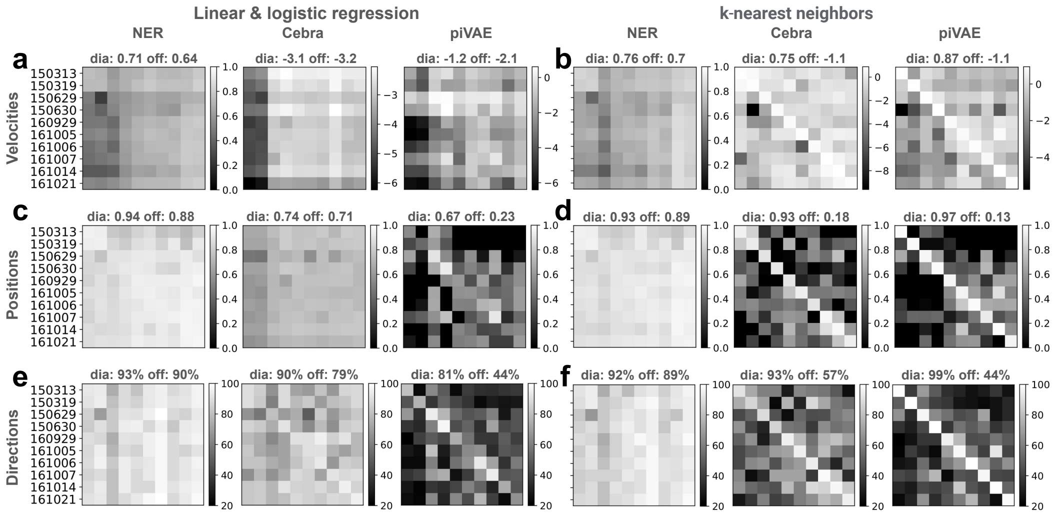

Figure 4: Decoding within and across time and brain hemisphere over years in M1. **a-d** Three methods (NER, Cebra, and piVAE) are applied to ten sessions' neural dynamics. Linear and logistic regression decoder **a, c, e** or nonlinear k-nearest neighbors decoder **b, d, f** are trained on 80% of data and are used to decode velocities, positions and directions on 20% remaining data within (diagonal) and cross sessions (off-diagonal). Brighter color indicates higher decoding performance. **a** Same session, cross session and hemisphere velocities decoding using linear regression decoder. Notice color bars are different for three methods and most values in Cebra and piVAE are negative. **b** Similar to **a** but using the nonlinear kNN decoder. The latent dynamics of Cebra is sixteen dimensions instead of three. **c-d** Similar to **a-b** but the dependent decoding variable is hand positions. Notice the range of color bars is 0-1 for all six figures. **e-f** Similar to **a-b** but the dependent decoding variable is hand directions. Notice the range of color bars is 0-100 for all six figures.

findings in the PMd (89% vs 32%) and S1 (86% vs 47%). Together, combined with linear decoders, NER shows clearest velocities dependent direction tuning and explains the largest variance in the velocities and positions.

2.3 Long term and cross hemispheres decoding in M1

NER explains the largest variance in both hand velocities and positions in the 80% training datasets, we next test the trained model on the remaining 20% of test data. In addition to decode test data from the same session of one day, we also test it against other sessions. We will perform the comparison in M1 first in the part (Fig 4). Then, we will turn our focus to PMd in the second part (Fig 5). Finally, we will compare it in the S1 (Fig 6). In addition to linear regression, we also tested a nonlinear decoder kNN.

Fig 4a shows the velocity decoding performance using linear regression decoder and three dimensionality reduction methods. Surprisingly, a linear decoder could not decode hand velocities using latent dynamics revealed Cebra (all variances are negative) and piVAE (only four positive). In contrast, all the variance in NER are positive (minimum: 0.24) and performance on different sessions is similar to within sessions (0.64 vs 0.71). A kNN decoder (Fig 4b) achieved high performance for Cebra and piVAE only for the within session latent dynamics. NER outperforms Cebra and piVAE by a large margin for all conditions in position decoding (Fig 4c). The kNN decoder does not improve NER performance compared to the linear decoder, but for Cebra and piVAE, the kNN decoder improves within session data performance over NER (Fig 4d). However, this sacrifices the cross session decoding performance (0.89 vs 0.18 and 0.13). Fig 4e shows the direction decoding accuracy where NER still outperforms Cebra and piVAE. We observed similar results using a kNN decoder (Fig 4f). Together, comparing with Cebra and piVAE, NER achieves much higher performance in all sessions using linear decoder and in cross sessions using nonlinear decoder.

Acknowledgments and Disclosure of Funding

We greatly appreciate the Miller and Shenoy labs for publicly releasing their experimental data on macaque monkeys. We also thank the Mathis lab for the CEBRA code, which served as the basis for our NER. This work was supported by National Institute of Deafness and Other Communications Disorders grant DC003180 (X.W.). No conflicts of interest are declared by the authors.

References

[1] Mehdi Azabou, Mohammad Gheshlaghi Azar, Ran Liu, Chi-Heng Lin, Erik C Johnson, Kiran Bhaskaran-Nair, Max Dabagia, Bernardo Avila-Pires, Lindsey Kitchell, Keith B Hengen, et al. Mine your own view: Self-supervised learning through across-sample prediction. *arXiv preprint arXiv:2102.10106*, 2021.

[2] Pinglei Bao, Liang She, Mason McGill, and Doris Y Tsao. A map of object space in primate inferotemporal cortex. *Nature*, 583(7814):103–108, 2020.

[3] Matteo Carandini and David J Heeger. Normalization as a canonical neural computation. *Nature reviews neuroscience*, 13(1):51–62, 2012.

[4] Le Chang and Doris Y Tsao. The code for facial identity in the primate brain. *Cell*, 169(6):1013–1028, 2017.

[5] Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012.

[6] Juan A Gallego, Matthew G Perich, Raeed H Chowdhury, Sara A Solla, and Lee E Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature neuroscience*, 23(2):260–270, 2020.

[7] Juan A Gallego, Matthew G Perich, Lee E Miller, and Sara A Solla. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, 2017.

[8] Juan A Gallego, Matthew G Perich, Stephanie N Naufel, Christian Ethier, Sara A Solla, and Lee E Miller. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature communications*, 9(1):4233, 2018.

[9] Cecilia Gallego-Carracedo, Matthew G Perich, Raeed H Chowdhury, Lee E Miller, and Juan Álvaro Gallego. Local field potentials reflect cortical population dynamics in a region-specific and frequency-dependent manner. *Elife*, 11:e73155, 2022.

[10] Joshua I Glaser, Ari S Benjamin, Raeed H Chowdhury, Matthew G Perich, Lee E Miller, and Konrad P Kording. Machine learning for neural decoding. *Eneuro*, 7(4), 2020.

[11] Cole Hurwitz, Akash Srivastava, Kai Xu, Justin Jude, Matthew Perich, Lee Miller, and Matthias Hennig. Targeted neural dynamical modeling. *Advances in Neural Information Processing Systems*, 34:29379–29392, 2021.

[12] Mohammad Reza Keshtkaran, Andrew R Sedler, Raeed H Chowdhury, Raghav Tandon, Diya Basrai, Sarah L Nguyen, Hansem Sohn, Mehrdad Jazayeri, Lee E Miller, and Chethan Pandarinath. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods*, 19(12):1572–1577, 2022.

[13] Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Xue-Lian Qi, Ranulfo Romo, Naoshige Uchida, and Christian K Machens. Demixed principal component analysis of neural population data. *elife*, 5:e10989, 2016.

[14] Eric Kenji Lee, Hymavathy Balasubramanian, Alexandra Tsolias, Stephanie Udochukwu Anakwe, Maria Medalla, Krishna V Shenoy, and Chandramouli Chandrasekaran. Non-linear dimensionality reduction on extracellular waveforms reveals cell type diversity in premotor cortex. *Elife*, 10:e67490, 2021.

[15] Ran Liu, Mehdi Azabou, Max Dabagia, Chi-Heng Lin, Mohammad Gheshlaghi Azar, Keith Hengen, Michal Valko, and Eva Dyer. Drop, swap, and generate: A self-supervised approach for generating neural activity. *Advances in neural information processing systems*, 34:10587–10599, 2021.

[16] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[17] Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

[18] Felix Pei, Joel Ye, David Zoltowski, Anqi Wu, Raeed H Chowdhury, Hansem Sohn, Joseph E O'Doherty, Krishna V Shenoy, Matthew T Kaufman, Mark Churchland, et al. Neural latents benchmark'21: evaluating latent variable models of neural population activity. *arXiv preprint arXiv:2109.04463*, 2021.

[19] Neil C Rabinowitz, Ben DB Willmore, Jan WH Schnupp, and Andrew J King. Contrast gain control in auditory cortex. *Neuron*, 70(6):1178–1191, 2011.

[20] Mostafa Safaie, Joanna C Chang, Junchol Park, Lee E Miller, Joshua T Dudman, Matthew G Perich, and Juan A Gallego. Preserved neural dynamics across animals performing similar behaviour. *Nature*, 623(7988):765–771, 2023.

[21] Omid G Sani, Hamidreza Abbaspourazad, Yan T Wong, Bijan Pesaran, and Maryam M Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1):140–149, 2021.

[22] Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368, 2023.

[23] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[24] Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.

[25] Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: Learning continuous representations for regression. *Advances in Neural Information Processing Systems*, 36, 2024.

[26] Ding Zhou and Xue-Xin Wei. Learning identifiable and interpretable latent models of high-dimensional neural activity using pi-vae. *Advances in Neural Information Processing Systems*, 33:7234–7247, 2020.

[27] Feng Zhu, Harrison A Grier, Raghav Tandon, Changjia Cai, Anjali Agarwal, Andrea Giovannucci, Matthew T Kaufman, and Chethan Pandarinath. A deep learning framework for inference of single-trial neural population dynamics from calcium imaging with subframe temporal resolution. *Nature neuroscience*, 25(12):1724–1734, 2022.

A Appendix / supplemental material

A.1 Related work

There are at least five categories of dimensionality reduction methods: First, linear methods such as PCA, jPCA[5], demixed PCA (dPCA)[13], and preferential subspace identification (PSID)[21]. PCA captures the majority of the variance in the data, jPCA reveals the rotational dynamics of monkey reaching, dPCA further reveals task-related components, and PSID can extract latent dynamics that predict motion during reach versus return epochs. Second, nonlinear methods like uniform manifold approximation and projection (UMAP)[16] and t-distributed stochastic neighbor embedding (tSNE)[23] have been extensively used in biological data, such as revealing different neuron cell types[14]. These methods can reveal identities but often collapse temporal dynamics that resemble neural trajectories. UMAP with labels has been used for dimensionality reduction[22][26]. Third, generative methods using recurrent neural networks (RNNs) or Transformers, such as latent factor analysis via dynamical systems (LFADS)[17], AutoLFADS[12], RADICaL[27], and Neural Data Transformer (NDT)[24], have been shown to model single-trial variability in neural spiking activity better than PCA. These methods have restrictive explicit assumptions about the underlying statistics of the data. Fourth, label-guided generative methods using VAE, such as Poisson identifiable VAE (piVAE)[26], SwapVAE[15], and targeted neural dynamical modeling (TNDM)[11]. For example, piVAE explicitly uses labels (both discrete and continuous) to shape the embeddings. It reveals eight well separated latent dynamics in M1 but cannot align with movements trajectories in each direction. Fifth, most recently, contrastive learning methods have been introduced for learning robust and generalizable representations of neural population dynamics, such as Cebra[22] and Mine Your Own vieW (MYOW)[1]. For example, comparing to piVAE, AutoLFADS, and UMAP, Cebra shows the most identifiable latent dynamics correponding to different hand directions in the S1, although the trajectoris of latent dynamics are not correlated with movements.

To fairly evaluate our dimensionality reduction method against related methods, we chose six representative methods from various categories: PCA, dPCA, UMAP (with and without labels), piVAE, and Cebra. To eliminate bias originating from a single session in a single brain area, where piVAE and Cebra were previously tested, we conducted experiments in M1, PMd, and S1 across a total of twenty sessions.

A.2 Model

Cebra considers each embedding in a batch (say 3) as a discrete class. For an anchor, it contrasts with its augmented embedding as a positive pair and 3 randomly sampled embeddings as negative pairs. NER ranks 6 embeddings according to their continuous labels. Then it contrasts an anchor with its augmented or 1st embedding as a positive pair and the remaining 4 embeddings as negative pairs. NER does not stop here; it continues by contrasting the 2nd embedding as a positive pair and the remaining 3 embeddings as negative pairs. This process continues until all the embeddings have been positively contrasted with the anchor. NER learns a regression-aware representation that orders all embeddings in a batch.

We are motivated by the fact that Cebra treats continuous labels as many discrete classes, which cannot be well separated in low-dimensional space. These classes are also highly imbalanced, with many more near-zero classes. NER aims to solve the high dimensionality and class imbalance issues present in Cebra.

We used the same sampling and neural feature encoder as Cebra[22] to extract the neural embeddings. Here, we explain how the neural embeddings are aligned with movement ranks by minimizing the Rank and Contrast loss (RNC[25]). The labels for the neural embeddings are XY velocities and reaching directions. Any paired embedding can be positive or negative depending on the comparison of their label distance to all other pairs within each batch. The training objective is to rank each neural embedding according to their rank in the labels.

Given an anchor v_i which is the feature got from the decoder, $S_{i,j} := \{v_k | k, d(\tilde{y_i}, \tilde{y_k}) \geq d(\tilde{y_i}, \tilde{y_j})\}$ denotes the set of samples that are of higher ranks than v_j in terms of label distance w.r.t. v_i, where $d(\cdot, \cdot)$ is the distance measure between two labels (e.g. L_1 distance). Then the normalized likelihood of v_j given v_i and $S_{i,j}$ can be written as

$$P(v_j|v_i, S_{i,j}) = \frac{exp(sim(v_i, v_j)/\tau)}{\sum_{v_k \in S_{i,j}} exp(sim(v_i, v_k))/\tau}$$

where $sim(\cdot, \cdot)$ is the similarity measure between two feature embeddings and τ denotes the temperature. Per-sample RNC loss is defined as the average negative log-likelihood over all other samples in a given batch:

$$l_{RNC}^{(i)} = \frac{1}{2N-1} \sum_{j=1, j \neq i}^{2N} -log \frac{exp(sim(v_i, v_j)/\tau)}{\sum_{v_k \in S_{i,j}} exp(sim(v_i, v_k)/\tau)}$$

L_{RNC} is enumerating over all N samples in a batch as anchors to enforce the entire feature embeddings ordered according to their orders in the label space:

$$L_{RNC} = \frac{1}{N} \sum_{i=1}^{N} l_{RNC}^{(i)} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{N-1} \sum_{j=1, \neq i}^{N} -log \frac{exp(sim(v_i, v_j)/\tau)}{\sum_{v_k \in S_{i,j}} exp(sim(v_i, v_k))/\tau}$$

A.3 Latent dynamics in PMd and decoding between M1 and PMd

Next, we turned our attention to PMd. Surprisingly, NER revealed the same latent dynamics in this higher order motor areas (Fig 5a). Two other dimensionality reduction methods also found similar latent dynamics but were less consistent and not aligned with movements (Fig 13). Fig 5b shows the within and cross session velocities decoding using linear regression decoder (left) and kNN decoder (right). Consistent to M1, in the PMd, first, both Cebra and piVAE fail using linear regression where NER achieves similar performance in all conditions, regardless of within and cross sessions. Second, NER is robust to different decoder whereas Cebra and piVAE change from failed state to outperform NER for within session latent dynamics using kNN decoder. Lastly, even using the kNN decoder, the Cebra and piVAE still fail on cross session decoding, whereas NER has similar performance. We also examined the positions and directions decoding performance and found that NER still outperforms two other methods (Fig 5c, Fig 14). Together, NER reveals consistent latent dynamics in PMd and could be applied to decode movements between PMd and M1.

A.4 Same movements but different latent dynamics in S1

Lastly, we checked the latent dynamics and movement decoding in the S1. NER reveals consistent latent dynamics in the S1 (Fig 6a, b). After rotating the latent dynamics with reference to the target in Fig 2, they show consistent but different shape than the latent dynamics observed in M1 and PMd. Velocity decoding using a linear regression decoder only works when the latent dynamics are extracted by the NER (Fig 6c, d). Although three methods all work for the positions decoding, the NER outperforms Cebra and piVAE in all nine conditions (Fig 6e). Although S1 displays different latent dynamics from M1 and PMd, NER is still the best method for decoding movement both within and across sessions in S1.

A.5 Curved movements with wider latent dynamics in M1

Lastly, we run both NER and Cebra on a new experiment where a monkey performs both straight and curved hand movements in different directions with simultaneous neural recordings in the M1 (Fig 7a). We first examined the latent dynamics when monkey perform straight hand movements at six directions (Fig 7b). Surprisingly, we found that both single and averaged latent dynamics align with movements (Fig 7c). Furthermore, they display similar shape in previously shown latent dynamics in M1. Next, we chose three hand directions and each direction has both straight and curved hand movements (Fig 7e). When we trained both NER and Cebra on single directions, only NER revealed clearly separated latent dynamics corresponding to straight and curved hand movements (Fig 7f). The difference among two methods are even clear when they were trained on three directions together: NER revealed narrower latent dynamics for straight movements that were surrounded but latent dynamics formed by the curved movements. The explained variance of NER is also better than Cebra for three angels, especially on the combined angles (Fig 16c). Last, we chose an even

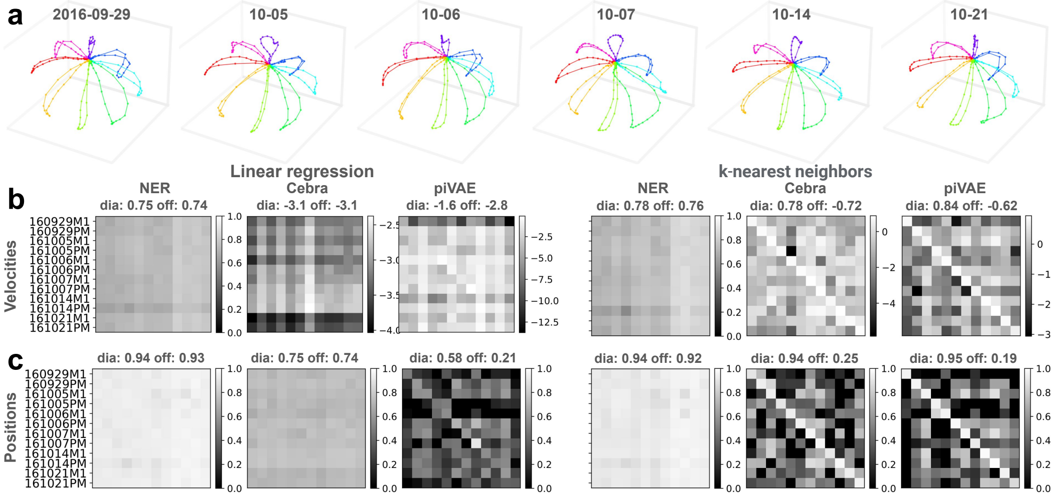

Figure 5: Latent dynamics in PMd and across brain areas decoding. **a** Trial averaged latent dynamics in the PMd revealed by NER. The rotation reference is same as Fig 2 (i.e., 2016-10-14). **b** Latent dynamics used previously in M1 are added. Same date, cross date and brain areas velocities decoding using linear regression (left) and kNN (right) decoder and latent dynamics reveals by three methods. Notice color bars are different for three dimensionality reduction methods and most values in Cebra and piVAE are negative. **c** Similar to **b** but the range of color bar is fixed (0-1). Fig 13 shows the single-trial and averaged latent dynamics revealed by Cebra and piVAE. Fig 14 shows the hand directions decoding performance using logistic regression decoder.

more challenging condition where all the six reaching movements are curved (Fig 7g). In the latent space, two curved movements at same direction have closer but separable latent dynamics (Fig 7h). Although Cebra archives compatible performance on single direction, it fails on combined directions (Fig 16e). NER also has higher explained variance than Cebra constantly (Fig 16f). Together, NER could not only align latent dynamics with straight movements, but also works well for aligning curved hand movements and differing them from straight hand movements.

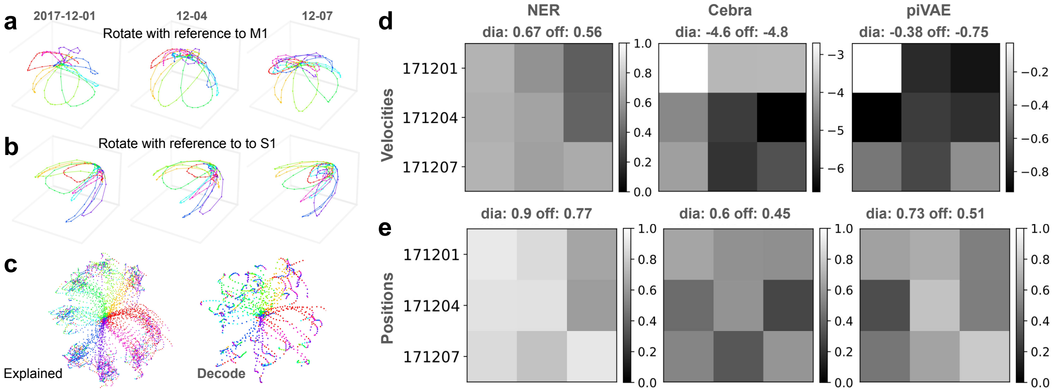

Figure 6: Distinct latent dynamics with same movement in S1. **a** Trial averaged latent dynamics revealed by NER. The reference target is same session used in Fig 2. **b** The reference target is first session in S1. **c** Left, on 80% held-in trials, the explained variance by linear (velocities) and logistic (for direction) regression decoders are 91% and 97%, respectively. Right, on 20% held-out test trials, trained linear decoder decode velocities and positions with performance of 71% and 90%, respectively. The data is collected on 2017-12-01. **d** Same and cross date velocities decoding performance using the linear regression decoder and three dimensionality reduction methods. Notice the color bars are different and all the values in the Cebra and piVAE are negative. **e** Similar to **d** but for positions. Notice the color bars are fixed for all three methods. Fig 15 shows latency dynamics revealed by Cebra and piVAE.

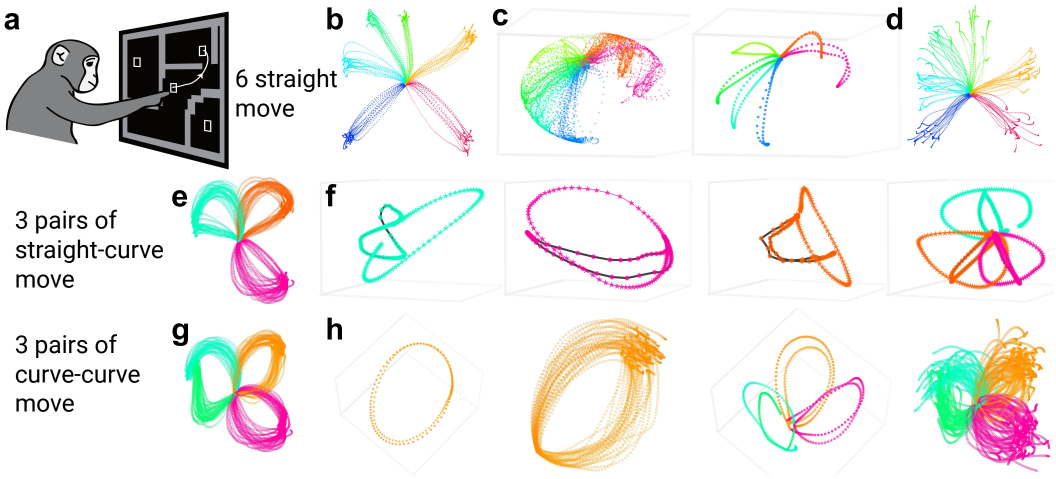

Figure 7: Distinct latent dynamics align with different movements in M1. **a** A monkey makes a curved reaching task through a virtual maze and neural activities from M1 are recorded simultaneously. There are multiple reaching directions and monkey need to make curved movements when there is a barrier on the trajectory. The cartoon from [18]. **b** Hand positions at six different target directions without barriers (i.e., straight hand movements). **c** Latent dynamics from single trial (left) and averaged (right). **d** Explained variance of hand movements only (velocity: 0.79, position: 0.92). The hand directions (i.e., colors) are assigned manually. **e** Hand movements at three different target directions without (straight movements) and with barriers (curve movements). **f** Latent dynamics of curved (stars) and straight (dots with black line) hand movements at each and combined directions together. **g** Hand movements at four different target directions that all have barriers (curve movements). **h** Trained latent dynamics from one pair (1st) or three pairs (3rd) of curved movements, and decoder explained hand movements (2nd and 4th). Fig 16 shows comparing results by Cebra.

A.6 Discussion

A benchmark of NER and six other dimensionality reduction methods across multiple brain areas and two movement tasks reveals its superior performance in uncovering latent dynamics. We believe the biggest advantage of our method is its ability to extract nearly identical latent dynamics across brain areas and over extended periods. This opens new avenues for both fundamental neuroscience research and brain-machine interfaces (BMI). [6][20] discovered preserved latent dynamics across time and animals performing similar behaviors using the PCA method. The latent dynamics revealed by NER in Fig 2, 5, 6, 7 are significantly more informative than those uncovered by PCA. We believe NER will help neuroscientists probe the stability of latent dynamics under various conditions. For BMI applications, we show that NER combined with a simple linear decoder can predict hand movements across years, brain areas, and hemispheres. This capability allows training latent dynamics within and between subjects and predicting movements in other subjects. The linear decoder's lack of hyperparameters is also a significant advantage.

The application of NER is not limited to hand movements using neurophysiological recordings. Latent dynamics can be revealed as long as the learning target is continuous. This includes latent dynamics in the hippocampus representing the body position of running rats and latent dynamics in the visual cortex representing embedded video features [22]. Similarly, the recording modalities are not limited to single-neuron electrophysiology; other methods such as calcium imaging, local-field potentials, and EEG can also be used. However, one limitation of our method, like others, is its ability to reveal latent dynamics only for stereotyped simple movement tasks. Each reaching direction needs to be repeated multiple times to obtain enough training samples. In daily life, natural movements are rarely stereotyped. Therefore, our method cannot be applied to random reaching tasks without binning trials with similar reaching directions[18]. Even when the reaching direction is repeated, our method (like others) cannot reveal identifiable latent dynamics representing more than 20 conditions. In our final experiments using curved movements, we manually chose three pairs of reaching tasks. Both NER and Cebra failed when all 108 conditions were trained simultaneously. More work is needed to address complex movements.

A.7 Datasets Code

Computer information: Operating System, Ubuntu, GPU, NVIDIA RTX A5000, computer memory, 42 GB

A.7.1 Center-out reaching experiments in M1, PMd, and S1

Behavior

The dataset comprises behavioral task data from two male Macaca mulatta monkeys (Monkeys H and C). These monkeys were trained to sit in a primate chair and perform a center-out reaching task using a planar manipulandum with the hand contralateral to the implanted hemisphere. During each trial, the monkey started by moving a cursor to a central target. After a variable waiting period, one of eight outer targets (equally spaced along a circle of 6–8 cm radius) was presented. Monkeys C and H differed in the task protocols:

Monkey C: Trained to wait for an auditory go cue during a delay period of 0.5–1.5 seconds while the target remained visible. Upon receiving the cue, the monkey had to move the cursor to the outer target within 1 second and hold it there for 0.5 seconds to receive a liquid reward.

Monkey H: No delay period; the monkey had to move the cursor to the outer target within 1 second and hold it there for 0.1 seconds.

For both monkeys, the trial restarted by returning the cursor to the central target. Endpoint positions of the manipulandum were recorded at 1 kHz, and task event timings were digitally logged. Hand velocity was computed as the derivative of hand position. The dataset includes 6 sessions for Monkey C and 5 sessions for Monkey H, considering only successful trials (an average of 307±221 trials per session, mean ± s.d.)

Neural recordings

The dataset consists of neural recordings from two male Macaque monkeys. These recordings were obtained using 96-channel Utah electrode arrays implanted in specific cortical regions.

Monkey C: Initially implanted in the right primary motor cortex (M1) and later received implants in the left M1 and dorsal premotor cortex (PMd) (denoted as CR and CL, respectively).

Monkey H: Implanted in area 2 of the primary somatosensory cortex of the left hemisphere.

Neural activity was recorded using a Cerebus system (Blackrock Microsystems, Salt Lake City, UT) at a sampling frequency of 30 kHz. The recorded signals underwent band-pass filtering (250–5000 Hz) and were converted to spike times based on threshold crossings. Spike sorting was performed using specialized software (Offline Sorter v3, Plexon, Inc, Dallas, TX) to identify putative neurons.

Date	Monkey	Hemisphere	Trial	M1	PMd	S1
150313	Chewie	Right	1038	86	n/a	n/a
150309	Chewie	Right	1026	72	n/a	n/a
150629	Chewie	Right	179	49	n/a	n/a
150630	Chewie	Right	178	44	n/a	n/a
160929	Chewie	Left	208	74	114	n/a
161005	Chewie	Left	202	82	167	n/a
161006	Chewie	Left	209	63	192	n/a
161007	Chewie	Left	168	70	137	n/a
161014	Chewie	Left	740	88	190	n/a
161021	Chewie	Left	286	84	211	n/a
171201	Han	Left	292	n/a	n/a	70
171204	Han	Left	255	n/a	n/a	83
171207	Han	Left	245	n/a	n/a	72

Table 1: Datasets for the center-out reaching experiments.

Datasets

All the center-out reaching experiments using the open source data from: `https://datadryad.org/stash/dataset/doi:10.5061/dryad.xd2547dkt` This data is released accompanying this paper:`https://elifesciences.org/articles/73155#data` We used all sessions from Monkey Chewie and Monkey Han. We chose these Monkeys because one session in Chewie is used by piVAE paper[26] and one session in Monkey Han is used by CEBRA paper[22]. Although the datasets come from same session in same Monkey, the temporal resolution is much higher for the datasets used by piVAE and CEBRA papers. The data is Matlab format and we extract following information: tgtDir (Target direction, radians for Monkey Chewie and degrees for Monkey Han), idx-goCueTime (The time go Cue is one), vel(XY velocities), M1-spikes for both Chewie 2015 and Chewie 2016, and PMd-spikes only for Chewie 2016. The time bin is 30ms and we extract all the spikes after each go Cue. We extracted 40 bins for Monkey Chewie and 35 bins for Monkey Han, because most trial in Monkey Han has short acquisition window than 40 bins (afte go Cue). We smoothed the discrete spike count in the Matlab using a Gaussian kernel. The standard deviation is 1.5 and kernel size is six standard deviations. We keep all the trials and neurons. The number of trials and neurons are shown in Table 1. Our NER is just a modification of the loss function used by CEBRA: `https://github.com/AdaptiveMotorControlLab/cebra` The RNC loss could be downloaded from: `https://github.com/kaiwenzha/Rank-N-Contrast` The iterations is 20000, learning rate is 1e-4, and batch size 512. The temperature is fixed to 1 for both NER and CEBRA. The output dimension of NER is fixed to 3. For CEBRA, we used output dimension of 3 for visualizing the latent dynamics and linear models decoding. We only used 16 dimensional embeddings for k-NN decoders. For the piVAE, we did not use the original version which is based on older version of Tensorflow. `https://github.com/zhd96/pi-vae` Instead, we used the modified conv-pi-VAE that is already included into the CEBRA package. We fixed the random seed to 42, and using batch size of 200 and iterations of 300.

A.7.2 Curved hand movements experiments in M1 Fig 7 16

The MC_Maze dataset includes recordings from the primary motor and dorsal premotor cortices of a monkey performing reaches to visual targets in a virtual maze with an instructed delay. This dataset comprises 108 different task configurations, each varying in target positions, barrier numbers, and barrier positions. The monkey repeated each task configuration multiple times in random order, resulting in 2,869 trials recorded in a single session with 182 neurons and simultaneous hand kinematics monitoring. This datasets could be downloaded from: `https://dandiarchive.org/dandiset/000128`, `https://github.com/dandisets/000128`. In our works, we used the NLB21 package[18] to download the data from DANDI: `https://github.com/neurallatents` Our code (Jupyter Notebook) is modified from: `https://github.com/neurallatents/neurallatents.github.io/blob/master/notebooks/mc_maze.ipynb` We used a time bin of 5ms (raw resolution is 1ms) and Gaussian window of 50ms. For six straight movements, we only use version 0 and trial type of 13, 29, 17, 38, 6, 18. For the straight-curved and curved-curved movements, we keep all three versions of task (one straight and two curved). We removed the target angle during training. We used trial type of 13, 38, 18 for straight-curved and 37,1, 31,38, 34,18 for curved-curved movements. The iterations is 5000, learning rate is 1e-4, and batch size 512.

A.8 Decoders

We rotate the latent dynamics with reference to same target before decoding. We used orthogonal Procrustes from scipy for this purpose: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.orthogonal_procrustes.html`. We picked on target angle and rotate the whole 3D latent dynamics using the computed orthogonal matrix. This rotation will not modify local detail or the relative positions of each reaching direction. For the linear decoders using linear and logistic regression, both are imported from "linear_model" of sklearn. There is no parameter or hyperparameter for the linear regression model. For the logistic regression model, we used the following parameters: max_iter=500, multi_class='multinomial', solver='lbfgs' For the k-nearest neighbors Regressor and Classifier, they are both imported from "neighbors" in sklearn. We used "GridSearchCV" in sklearn to searach the best "n_neighbors" range from 2 to 1024.

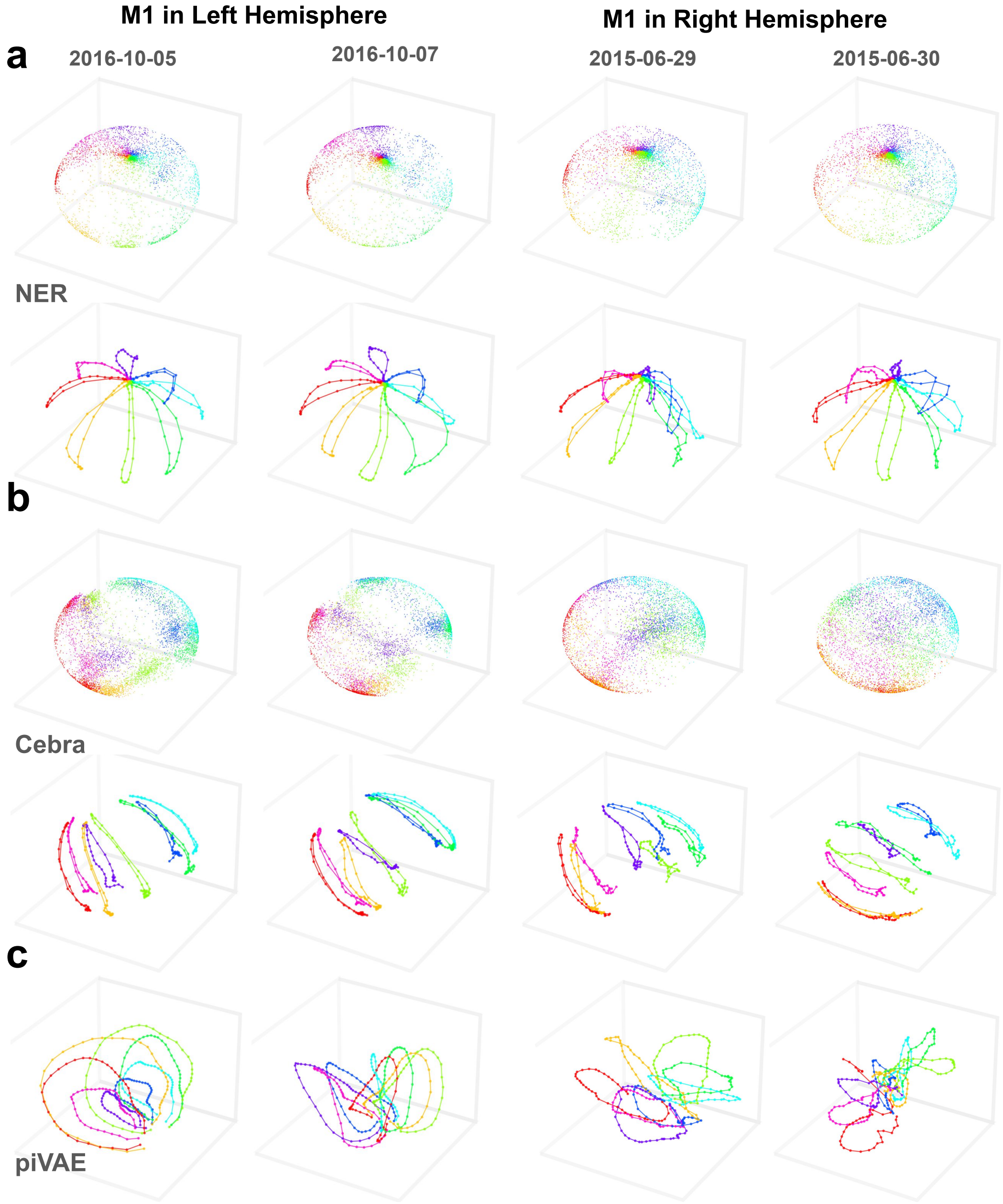

Figure 8: NER reveals consistent and movements aligned latent dynamics in M1 for the remaining four sessions. Extra four sessions' latent dynamics at left and right hemisphere of Monkey C after rotating relative to target session in Fig 2 (2016-10-14).

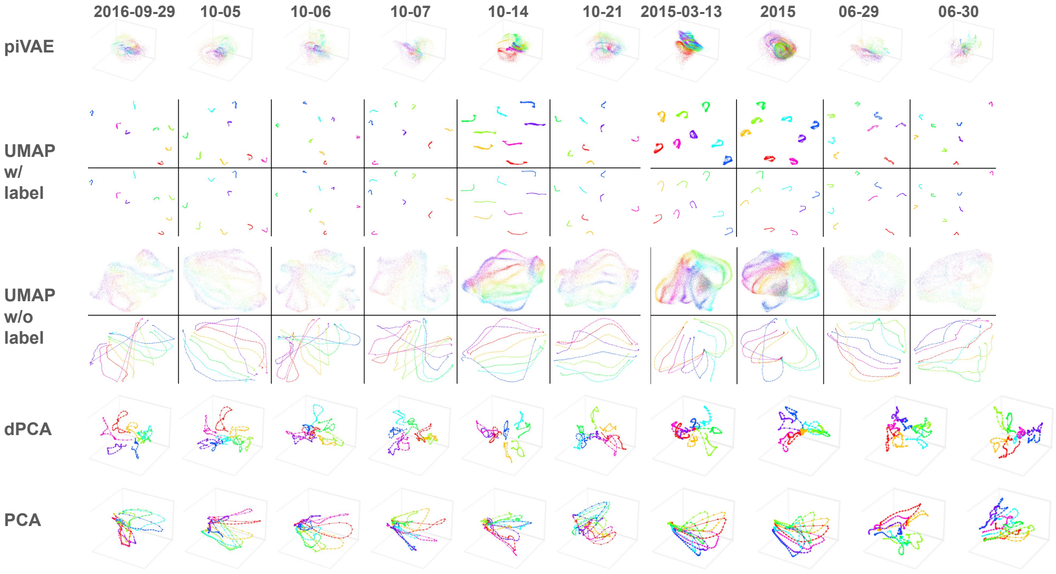

Figure 9: Neural embeddings revealed by five other dimensionality reduction methods. Single trial (top) and trial averaged (bottom) latent dynamics revealed by five other dimensionality reduction methods.

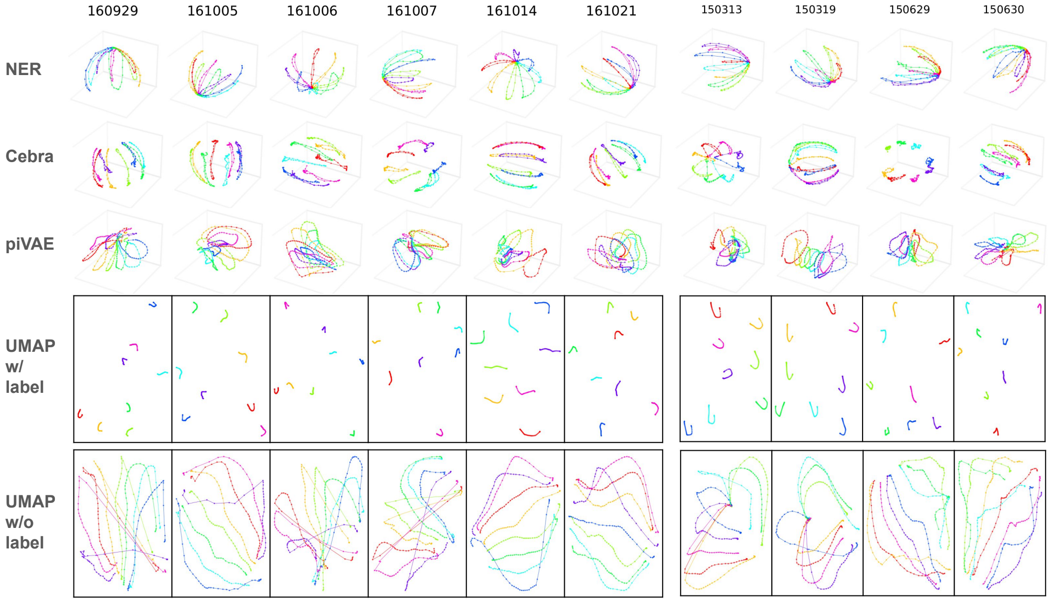

Figure 10: Latent dynamics without rotation. Trial-averaged latent dynamics revealed by five dimensionality reduction methods without rotation.

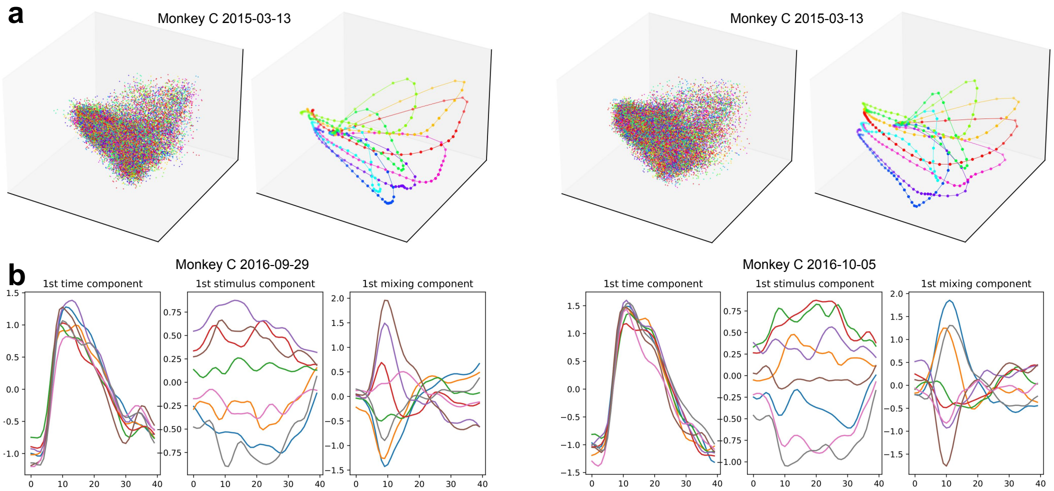

Figure 11: Mixed single trial latent dynamics in PCA and time-stimulus components revealed by dPCA. **a** Unlike other five dimensionality reduction methods, single trial latent dynamics revealed by principal component analysis (PCA) and demixed PCA (dPCA) is mixed and latency dynamics are only identifiable after averaging. **b** In the trial-averaged latent dynamics, dPCA reveals three components (left, middle, right) at eight directions (different colors): time component aligned with go cue regardless of directions, separated stimulus component varied across time, and mixed component aligned with go cue and different for each direction.

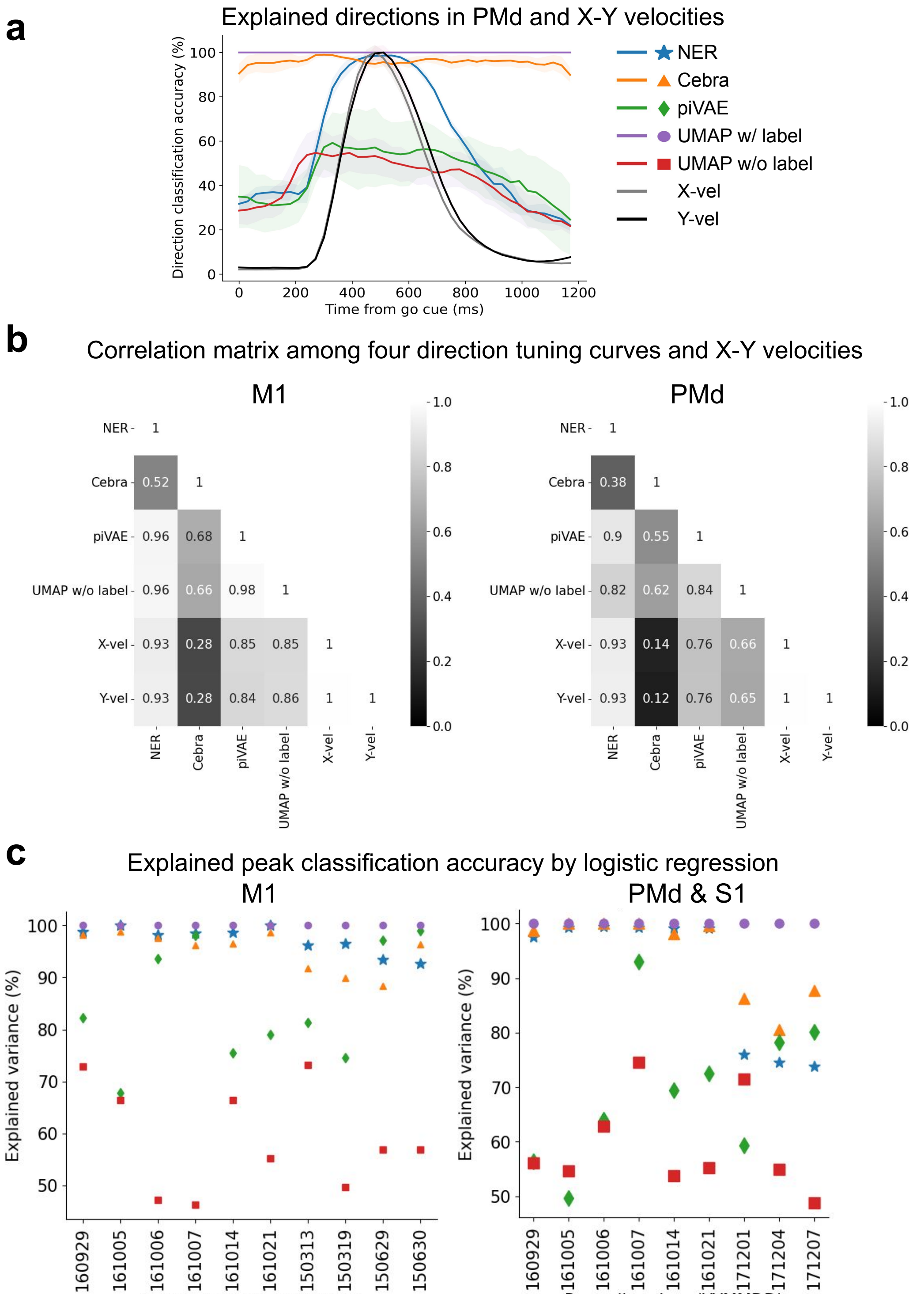

Figure 12: Hand directions tunings and explained peak classification accuracy of hand directions in M1, PMd, and S1. **a** Hand directions explained accuracy using a logistic regression models trained on the latent dynamics revealed by five dimensionality reduction methods. Shaded areas are standard deviation over six sessions from PMd in the Monkey C. Notice the only NER reveal hand velocities dependent direction tuning curves that peak around 500 ms. **b** Correlation coefficients matrix between direction tuning curves and velocities in M1 (left) and PMd (right). **c** Explained variance on hand directions using logistic regression decoder.

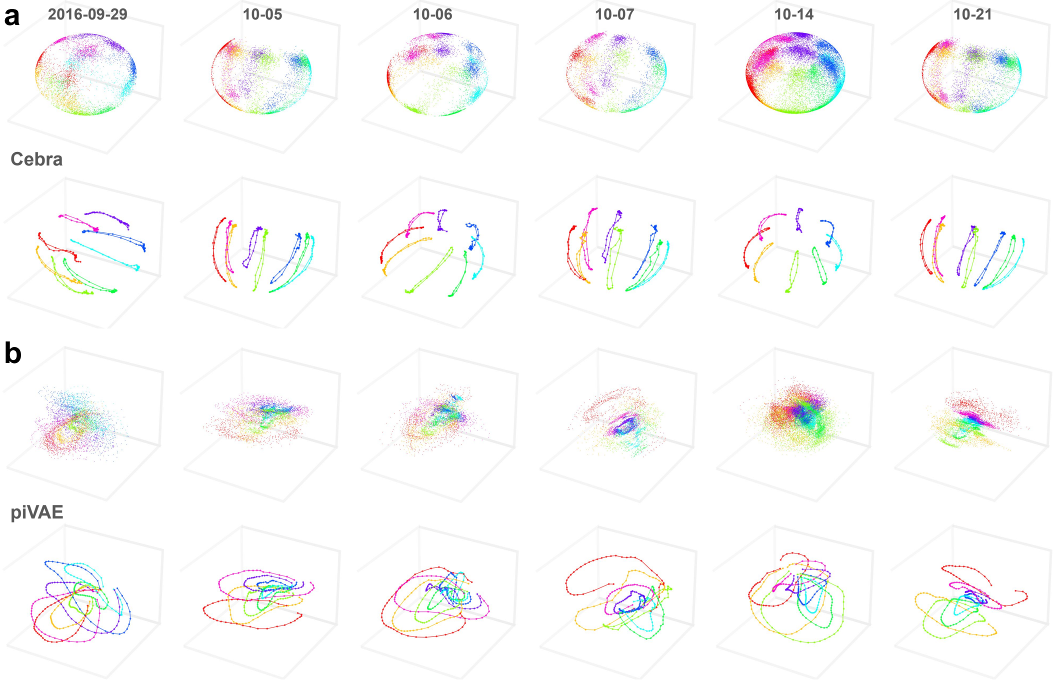

Figure 13: Latent dynamics in PMd revealed by Cebra and piVAE. Single trial and trial averaged latent dynamics revealed by Cebra and piVAE. All the figures are rotated with reference with the one session shown in Fig 2 (2016-10-14).

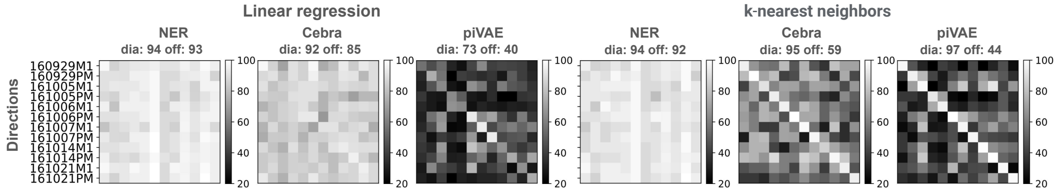

Figure 14: Direction decoding accuracy. Same date, cross date, and cross brain areas decoding of hand directions using linear regression (left) and k-nearest neighbors decoder trained on the latent dynamics revealed by three methods. Notice the range of color bars is 0-100 for all six figures.

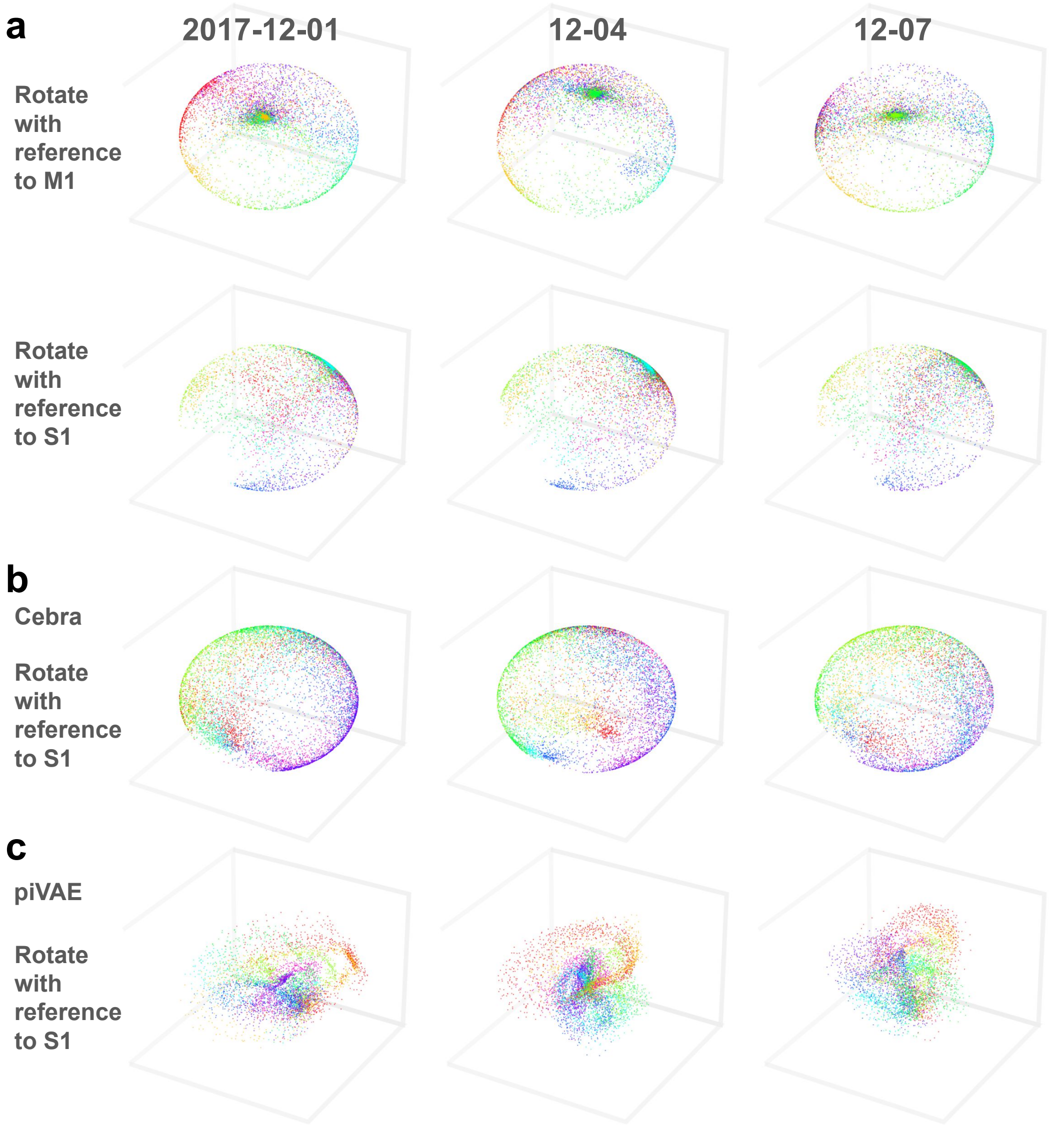

Figure 15: Latent dynamics in S1 revealed by Cebra and piVAE. **a** Top, single trial latent dynamics revealed by Cebra are rotated with reference to the one session in M1. Bottom, latent dynamis are rotated with reference to the first session in S1. **b** Similar to the bottom figure of a but using Cebra dimensionality reduction method. **c** Similar to c but using piVAE dimensionality reduction method.

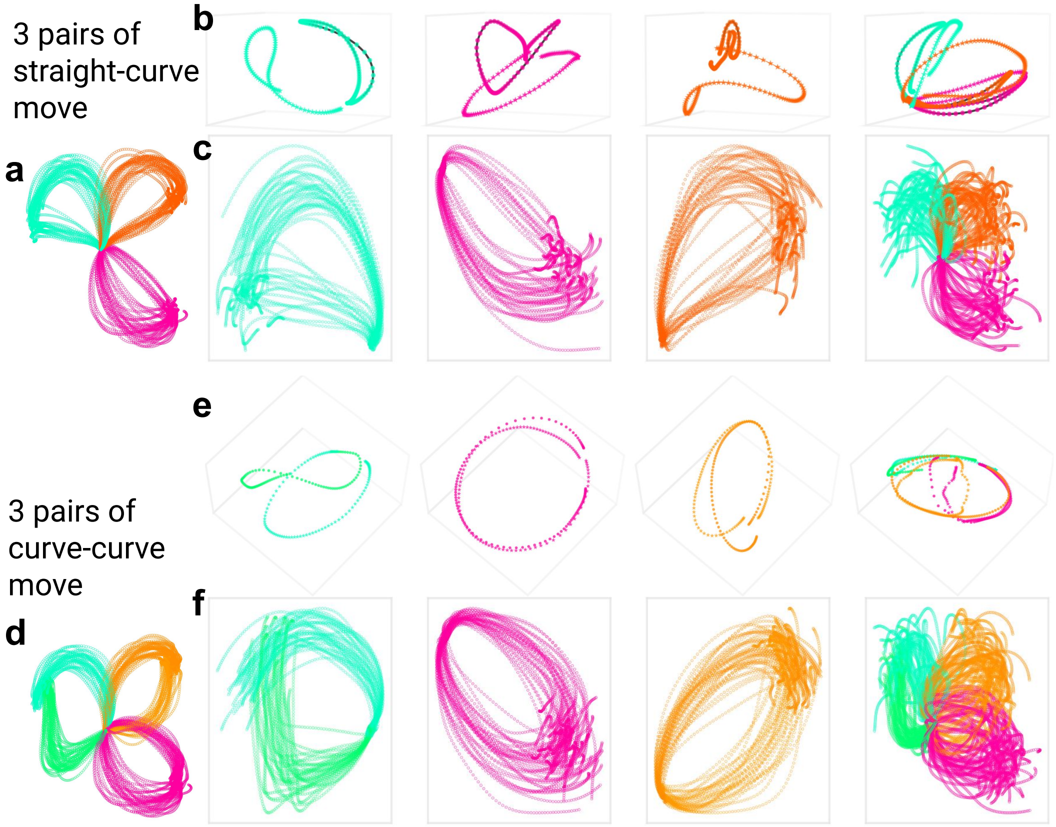

Figure 16: Distinct latent dynamics revealed by Cebra. Similar to Fig 7 but using the latent dynamics revealed by Cebra. **a** Ground truth motion trajectories at three directions for straight-curve movements. **b** Latency dynamics trained on three directions separately and combined. Notice the latent dynamics of straight hand movements (dots and black line) are either mixed (1st) with curved movements or squeezed (3rd). Latent dynamics trained by three directions combined are overlapped. **c** Linear models predicted hand trajectories. The directions are manually assigned. The explained variance for hand velocities (directions) are 87% (94%), 83% (93%), 82% (94%), and 66% (86%). Notice all the values are lower than NER which are 92% (95%), 89% (96%), 90% (96%), and 81% (91%), especially for the combined directions. **d** Ground truth motion trajectories at four directions for paired curve-curve movements. **e** Latency dynamics trained on three directions separately and combined. Notice latent dynamics trained by three directions combined are overlapped. **f** Linear models predicted hand trajectories. The directions are manually assigned. The explained variance for hand velocities (directions) are 85% (93%), 81% (92%), 85% (94%), and 61% (80%). Notice all the values are lower than NER which are 87% (95%), 89% (95%), 91% (96%), and 81% (91%), especially for the combined directions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims we made in the abstract and introduction are based on our extensive experiments over multiple dataset and a benchmark comparisons of our method with six other methods.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In the last part of Discussion, we mentioned two limitations of this work clearly

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: There is no theoretical analysis. This work an application of contrastive learning and deep regression on Neuroscience. We do not make or claim any theoretical contribution.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our works use the public dataset, and our code adds the loss function from one paper into the code of the other paper. Both codes are available online. There is not hyperparameter turnings. We mentioned the GPU, iterations, and batch size we used in the Appendix. We also upload our codes that generate all the Figures.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Also the data and code are already available online. We just combine two codes.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We gave those details in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: We just show the raw, mean and median values without using any statistical metrics.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the details of our computer that used to run those experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We used open-source neural spiking data from animals.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The data we used came from neuron activities in the brains. Nothing is related to humans.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The codes have been released by others already.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: All the data and codes are open source

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Everything is open source already.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Only authors do the experiments.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We used open source data.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.