DDAlign: A Transformer-based framework for long-term neural manifold alignment and stable cross-day decoding

Xinrui Yang and Tielin Zhang*

Center for Excellence in Brain Science and Intelligence Technology, Institute of Neuroscience, Chinese Academy of Sciences, Shanghai 200031, China.

ABSTRACT

Neural decoders often degrade across days due to slow drift in population activity. We introduce DDAlign, a transformer-based framework that stabilizes neural manifolds over long time periods. The model learns latent trajectories with an explicit velocity field and applies a lightweight day-specific aligner that corrects drift with minimal calibration. We evaluate DDAlign on two multi-week macaque datasets. It consistently achieves the highest cross-day decoding accuracy, preserves stable latent geometry, and maintains performance close to the within-day baseline. Ablation results show that both the velocity encoder and alignment head are essential for long-term robustness. DDAlign provides an efficient and general approach for stable neural decoding in practical brain–computer interface settings.

Keywords: Brain–computer interfaces, Neural manifold alignment, Cross-day decoding

1. INTRODUCTION

Brain computer interfaces rely on neural population activity to predict movement or force. Models trained on intracortical recordings can reach high decoding accuracy within a single day. However, the performance often becomes unstable across days. This instability is a major obstacle for long-term BCI use. Many biological and technical factors lead to daily changes in neural activity. These factors include electrode micromotion, neural adaptation and behavioral variability. As a result, the mapping between neural signals and behavior changes over time. Even small changes can cause a decoder trained on one day to perform poorly on the next day. Stable long-term decoding therefore requires methods that can handle cross day variability. This creates a need for a principled framework that can represent population activity in a stable way and link neural dynamics to behavior.

The concept of the neural manifold provides a principled framework for representing population activity. Although individual neurons exhibit high dimensional and noisy firing patterns, their collective activity often resides in a low dimensional latent space known as the neural manifold.[1,2] This latent space captures the dominant modes of population activity during specific tasks and reflects the organized structure of cortical representations.[3] Modeling neural activity within this manifold helps reduce noise while preserving task relevant geometry.[4] Importantly, the neural manifold is not stable across days. Population activity recorded on different days can occupy distinct regions of the latent space while maintaining similar overall shapes, indicating structured across day drift.[5,6] Such drift can arise from electrode micromotion, changes in neural excitability, and behavioral variability, and tends to follow consistent directions in population activity rather than random noise. Even small shifts in the manifold can disrupt the relationship between latent neural activity and behavior, making it essential to model and compensate for drift to achieve stable long term decoding.[6–8]

Neural activity is not only organized in space but also evolves over time. Population activity traces smooth trajectories on the neural manifold, reflecting the internal dynamics of the motor system. Several recent models have leveraged this structure. LFADS[4] infers smooth latent trajectories from neural activity, CEBRA[9] learns continuous representations that preserve temporal neighborhoods and behavioral similarity, and MARBLE models latent dynamics to predict future neural states. These approaches demonstrate that neural dynamics contain

Further author information: (Send correspondence to Tielin Zhang.)
Tielin Zhang: E-mail: zhangtielin@ion.ac.cn

meaningful information beyond static latent position. In particular, changes in the latent state over time provide insight into how neural trajectories evolve on the manifold. If the latent state at time t is Z_t, the difference $Z_{t+1} - Z_t$ describes the local direction of movement along the trajectory. This change can be interpreted as a latent velocity, which provides a compact description of neural dynamics. This representation is especially relevant for motor decoding, as behavior depends on both state and movement. When neural activity drifts across days, changes can occur in both the manifold geometry and its temporal evolution. Addressing long term decoding stability therefore requires modeling both geometric drift and alterations in latent dynamics. Modeling such long range temporal dependencies and their evolution across days requires a representation that can flexibly capture global temporal relationships in neural population activity, which motivates the use of attention based sequence models.

2. RELATED WORKS

Neural alignment has been studied from several perspectives. These methods aim to reduce variability across days, subjects, or tasks. Although they have shown promising results, each family has important limitations when applied to practical neural decoding. This section summarizes three main categories of alignment methods and discusses their strengths and weaknesses.

Geometric alignment methods. These approaches align neural activity using linear transforms such as Procrustes analysis and canonical correlation analysis.[6,8] They assume that neural manifolds across days share similar geometry, so alignment reduces to estimating an optimal rotation or linear mapping. These methods are simple, computationally efficient, and can work well when drift is small or labels are available. However, geometric methods rely on supervision and often require matched labels or paired trials.[6] Their linear transforms cannot capture nonlinear drift or local manifold deformation, which leads to degraded performance when drift accumulates over long periods or when recording conditions differ across subjects.[7,8]

Distribution alignment methods. These approaches match the statistical distributions of neural activity across days. Examples include adversarial domain adaptation networks and generative models such as CycleGAN.[10] NoMAD[11] follows this idea and introduces a distribution matching objective within a latent dynamical framework. These methods do not require labels, which is advantageous for neural datasets with limited annotations, and they can align sessions without trial correspondence. However, distribution alignment often requires large datasets to estimate stable statistics, while neural recordings are typically limited in size. Adversarial training can also introduce instability,[10] and the resulting models are often computationally expensive. These factors can limit their effectiveness in small sample decoding scenarios.

Dynamics based alignment methods. These approaches learn latent trajectories that capture the temporal structure of neural activity. Representative models include LFADS,[4] CEBRA,[9] and MARBLE.[12] They can learn smooth latent dynamics and often provide informative representations for decoding. By focusing on underlying temporal structure, these methods can be robust to changes in firing rate or noise, and many allow unsupervised training. However, these models are primarily designed to learn latent dynamics within individual sessions rather than explicitly enforcing correspondence across days. As a result, cross day alignment may emerge implicitly but is not directly optimized. In addition, they typically require long recordings for stable training, which can limit their applicability when data availability is restricted.

3. METHODS

3.1 Model Overview

The model converts a sequence of neural population activity into a two dimensional trajectory. It contains three components: an encoder that outputs a latent trajectory and a latent velocity field, a day-specific aligner that reduces cross day drift, and a decoder that maps the aligned latent states to behavior (Fig. 1). Neural features are z scored using training data, per day during alignment training, and using the first K trials of each test day.

Each trial provides neural observations

$$x(t) \in \mathbb{R}^D, \quad t = 1, \ldots, T, \tag{1}$$

and the goal is to predict

$$y(t) \in \mathbb{R}^2. \tag{2}$$

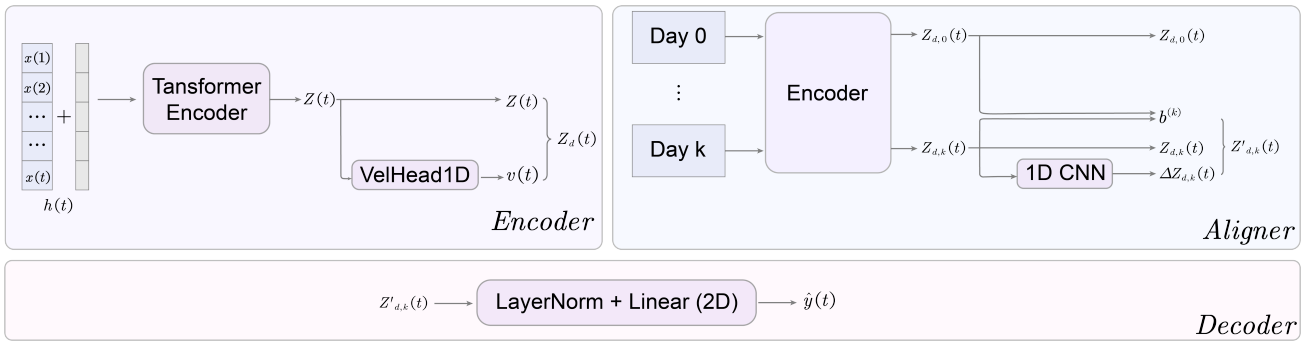

Figure 1. Model overview. The encoder outputs a latent trajectory $Z_d(t)$ and a latent velocity $v(t)$. A day-specific aligner produces aligned latent states $Z'_{d,k}(t)$ using a drift bias $b^{(k)}$ and a dynamic correction $\Delta Z_{d,k}(t)$, which are mapped by the decoder to the two dimensional prediction $\hat{y}(t)$.

3.2 Encoder: latent trajectory and latent velocity

The encoder maps the input sequence to a latent trajectory and a latent velocity field. We first project the neural input into a d dimensional embedding

$$h(t) = W_{\text{in}}x(t) + b_{\text{in}} + p(t), \tag{3}$$

where $p(t)$ is a positional encoding.

A Transformer encoder processes $\{h(t)\}_{t=1}^{T}$ and produces a latent state

$$z(t) = \text{Enc}\big(h(1), \ldots, h(T)\big)_t. \tag{4}$$

To represent local dynamics on the latent manifold we apply a temporal convolutional head, named `VelHead1D` in the figure. This module predicts a latent velocity field

$$v(t) = f_{\text{vel}}\big(z(1), \ldots, z(T)\big)_t. \tag{5}$$

We define a dynamic latent representation by an Euler style update

$$Z_d(t) = z(t) + v(t). \tag{6}$$

For a day k we denote the encoder output by $Z_{d,k}(t)$. We use three levels of latent variables. The transformer produces $z(t)$, the latent velocity field defines $Z_d(t) = z(t) + v(t)$, and the aligner maps $Z_{d,k}(t)$ to an aligned state $Z'_{d,k}(t)$. This separation ensures that dynamics and alignment operate on different stages of the representation.

The base model is trained with a trajectory MSE loss and two velocity consistency losses to stabilize the learned dynamics.

3.3 Day-specific aligner

For each non anchor day k, the aligner maps $Z_{d,k}(t)$ to the reference manifold.

Drift correction A day-specific bias models slow shifts:

$$Z^{\text{drift}}_{d,k}(t) = Z_{d,k}(t) + b^{(k)}. \tag{7}$$

Dynamic correction A shared temporal CNN models temporal deformation:

$$\Delta Z_k(t) = g_{\text{dyn}}\big(Z^{\text{drift}}_{d,k}(1), \ldots, Z^{\text{drift}}_{d,k}(T)\big)_t. \tag{8}$$

Aligned latent representation The final aligned state is

$$Z'_{d,k}(t) = Z^{\text{drift}}_{d,k}(t) + \alpha\, \Delta Z_k(t), \tag{9}$$

with α a small constant controlling the correction strength.

3.4 Decoder

A LayerNorm and linear layer map the aligned latent state to the predicted trajectory:

$$\hat{y}(t) = W_{\text{out}}\text{LN}\big(Z'_{d,k}(t)\big) + b_{\text{out}}. \tag{10}$$

3.5 Training the aligner

The encoder and decoder are frozen. For each training day k, we compute $Z_{d,k}(t)$, apply the aligner to obtain $Z'_{d,k}(t)$, and decode to $\hat{y}_k(t)$.

The supervised loss is

$$\mathcal{L}_{\text{pos}} = \frac{1}{NT}\sum_{n,t} \|\hat{y}_n(t) - y_n(t)\|_2^2 . \tag{11}$$

To limit alignment strength we penalize latent changes:

$$\mathcal{L}_{\Delta Z} = \frac{1}{NT}\sum_{n,t} \big\|Z_{d,k}(t)_n - Z'_{d,k}(t)_n\big\|_2^2 . \tag{12}$$

The aligner objective is

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{pos}} + \lambda_{\Delta Z}\mathcal{L}_{\Delta Z}. \tag{13}$$

3.6 Test time calibration

For each test day k, z score statistics are computed from the first K trials. At test time we keep the dynamic correction g_{dyn} fixed and only adapt the drift bias $b^{(k)}$ using an unsupervised objective,

$$\mathcal{L}_{\text{calib}} = \lambda_{\Delta Z}\mathcal{L}_{\Delta Z}, \tag{14}$$

computed on calibration trials. The updated $b^{(k)}$ is then applied to all trials of day k.

3.7 Summary

The encoder provides latent representations with explicit dynamics. The aligner applies day specific drift and dynamic corrections. At test time only a single parameter is adapted, enabling stable cross-day decoding without overfitting.

4. RESULTS

4.1 Data

We used two publicly released neural recording datasets from Ma et al.,[10] which include multi-week intracortical recordings collected from rhesus macaques implanted with 96-channel Utah arrays in primary motor cortex.

The first dataset contains recordings from monkey Jango performing an isometric wrist–force task. The monkey controlled a cursor by generating two-dimensional wrist forces toward one of eight targets. Neural activity was converted to smoothed firing rates in 50 ms bins. We used the first 32 trials of each day for day-specific normalization and evaluated decoding on the remaining trials. This dataset spans multiple weeks and serves as the main test bed for cross-day alignment. Fig. 2a illustrates the task paradigm.

The second dataset contains recordings from monkeys C and M performing a planar center-out reaching task. We used the smoothed firing rates and the two-dimensional cursor trajectories as regression targets. This dataset provides an additional validation set for evaluating cross-day generalization.

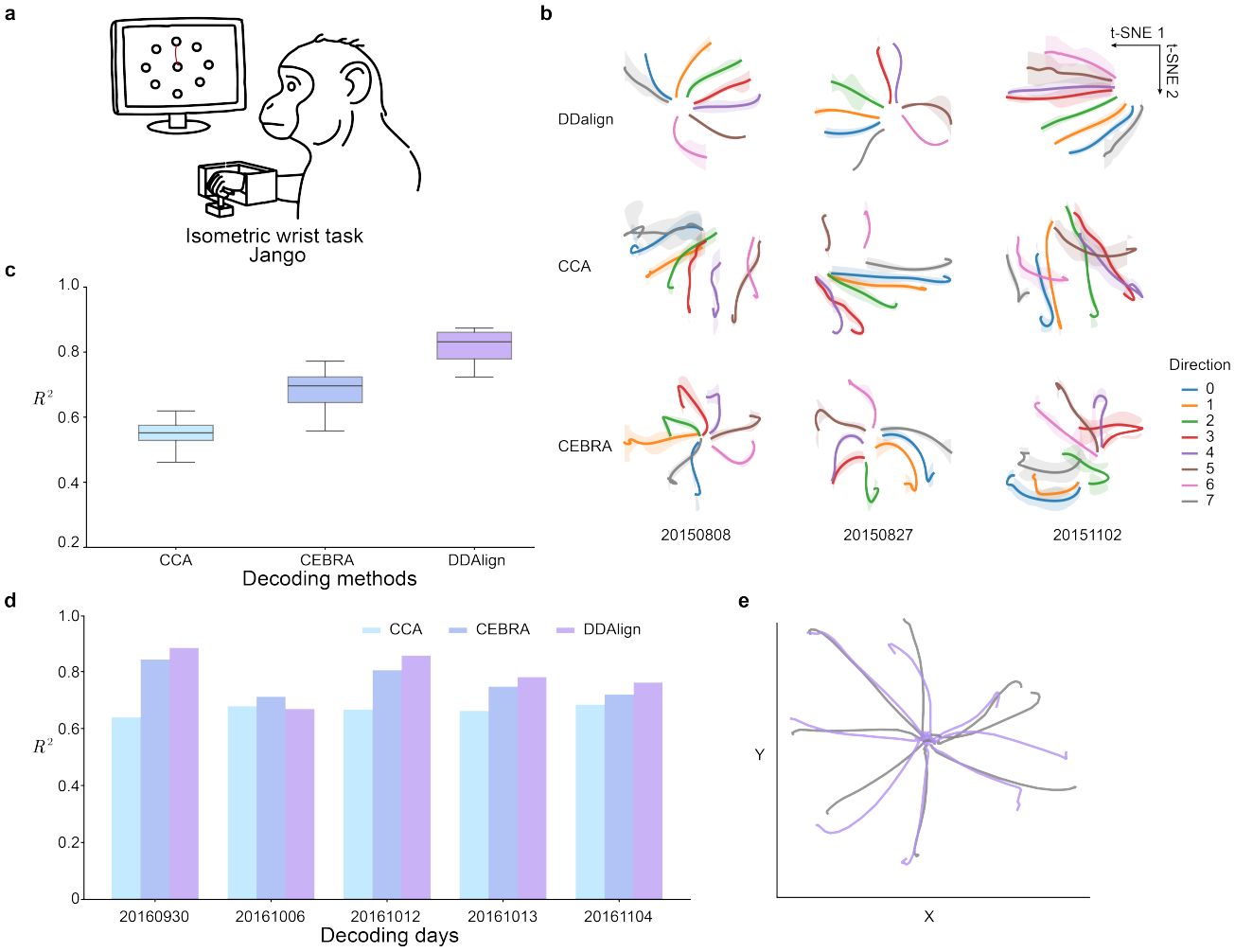

Figure 2. Cross-day decoding and latent manifold alignment. (a) Task paradigm for the Jango isometric wrist–force task. (b) Latent manifold visualization from three sessions (early, middle, late). (c) Cross-day decoding performance on the Jango dataset. (d) Cross-day decoding performance on the CM dataset. (e) Example reconstructed hand trajectories from the CM dataset.

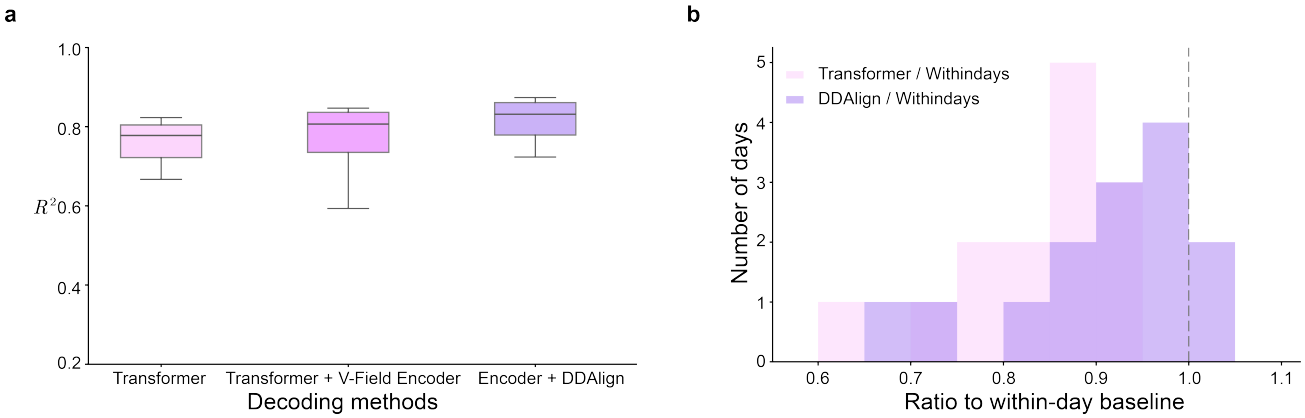

Figure 3. (a) Cross-day decoding performance for three model variants: the Transformer baseline, the Transformer with the velocity encoder, and the full DDAlign model. (b) Distribution of decoding ratios relative to within-day performance. DDAlign shows the highest and most concentrated ratios, indicating improved long-term stability.

4.2 Cross-day decoding performance

We evaluated cross-day decoding on two independent datasets. The first dataset contains fourteen test days from the Jango isometric-force task (Fig. 2c). The second dataset contains five additional days from the CM reaching task (Fig. 2d). We compared three alignment methods: CCA,[6] CEBRA[9] and DDAlign.

CCA was implemented following Gallego et al. (2020). We used 200 trials per test day for unsupervised alignment. Neural activity was reduced with PCA before computing the linear transform. CEBRA was reproduced using the official implementation. We used 32 calibration trials with pseudo-labels for supervised training, and a KNN decoder for prediction. DDAlign was trained only on the calibration trials and did not require paired data.

Across all fourteen Jango days, DDAlign achieved the highest R^2 values (mean around 0.83). CEBRA performed moderately well (mean around 0.72). CCA showed the lowest performance (mean around 0.55) and degraded on several days (Fig. 2c). Results on the CM dataset showed the same trend (Fig. 2d). DDAlign outperformed the other two methods on all five test days. This confirms that DDAlign provides more stable long-term alignment across different tasks and animals.

4.3 Latent manifold alignment

We examined the latent representations across a long recording period. Fig. 2b shows t-SNE embeddings from the first test day (20150808), a mid-range day (20150827) and the last test day (20151102). These results are from the Jango dataset. These three sessions span several weeks, and they represent the beginning, middle and end of the evaluation period. Each curve corresponds to one movement direction.

CCA produced unstable latent structures. Direction trajectories overlapped or changed shape across days. The same direction occupied different regions on early, middle and late sessions, indicating that linear alignment cannot maintain long-term manifold geometry. CEBRA formed clearer clusters on the first test day. However, the structure drifted on the mid and late sessions. Several directions became fragmented or rotated on 20150827 and 20151102. This shows that supervised embedding becomes unstable when only a small calibration set is available.

DDAlign produced the most consistent latent geometry over time. Direction trajectories were compact and well separated on all three sessions. Their shapes remained stable from the first to the last test day. Clusters from different days almost overlapped, showing minimal drift over the entire multi-week period. In addition, Fig. 2e shows reconstructed hand trajectories from the CM dataset, where DDAlign again preserves consistent directional structure and reduces across-day distortions. These results demonstrate that DDAlign preserves task-related neural structure and provides robust manifold alignment across long time intervals.

4.4 Analysis of Model Variants

We evaluated the contribution of each module using three variants: the Transformer baseline, the Transformer with the velocity encoder, and the full DDAlign model.

Fig. 3a shows that the velocity encoder improves performance across almost all fourteen days. The mean R^2 rises from around 0.76 to 0.80. The lowest days also become less severe. This indicates that latent velocity information helps stabilize the dynamics learned by the model. DDAlign achieves the best performance. Its mean R^2 reaches about 0.83 and remains high across all days. The model reduces errors on difficult sessions and preserves accuracy on easy ones.

Fig. 3b compares each method to the within-day baseline. Transformer ratios mostly fall between 0.70 and 0.90. Adding the velocity encoder shifts the ratios upward. DDAlign produces the tightest distribution, with most days between 0.90 and 0.95 and several near one. This shows that DDAlign maintains cross-day decoding close to the within-day level.

Together these results show that the velocity encoder improves decoding stability, and the alignment head further reduces long-term drift. Both modules are necessary for the full cross-day robustness of DDAlign.

5. CONCLUSION

Across all analyses, DDAlign provided the most stable and accurate cross-day decoding. The model uses a transformer backbone together with a velocity encoder and a lightweight alignment head. These components jointly preserve latent structure and reduce long-term drift, without requiring paired data. Experiments on the Jango and CM datasets showed that DDAlign maintained consistent manifold geometry over several weeks, preserved clear directional structure even on difficult recording days, and achieved the highest decoding accuracy among all methods. The model variants further demonstrated that the velocity encoder stabilized the latent dynamics, while the alignment head effectively corrected slow drift. Together they enabled cross-day performance close to within-day decoding.

These results indicate that explicitly modeling both latent dynamics and slow drift provides an effective and practical approach for long-term neural decoding. In contrast to methods that rely on implicit alignment through embedding or large scale distribution matching, DDAlign directly integrates dynamics modeling with lightweight calibration, making it suitable for settings with limited data. The DDAlign framework is general and can be extended to other recording modalities, tasks, and alignment scenarios. Future work will explore rapid adaptation with small samples, cross-subject alignment, and broader cross-system generalization to support more stable long-term brain–computer interfaces.

ACKNOWLEDGMENTS

This work was supported by projects 2024YFF1400600 and 2024YFF1400604.

REFERENCES

[1] Cunningham, J. P. and Yu, B. M., "Dimensionality reduction for large-scale neural recordings," *Nature neuroscience* **17**(11), 1500–1509 (2014).

[2] Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., and Shenoy, K. V., "Neural population dynamics during reaching," *Nature* **487**(7405), 51–56 (2012).

[3] Gallego, J. A., Perich, M. G., Miller, L. E., and Solla, S. A., "Neural manifolds for the control of movement," *Neuron* **94**(5), 978–984 (2017).

[4] Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R., et al., "Inferring single-trial neural population dynamics using sequential auto-encoders," *Nature methods* **15**(10), 805–815 (2018).

[5] Pun, T. K., Khoshnevis, M., Hosman, T., Wilson, G. H., Kapitonava, A., Kamdar, F., Henderson, J. M., Simeral, J. D., Vargas-Irwin, C. E., Harrison, M. T., et al., "Measuring instability in chronic human intracortical neural recordings towards stable, long-term brain-computer interfaces," *Communications Biology* **7**(1), 1363 (2024).

[6] Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A., and Miller, L. E., "Long-term stability of cortical population dynamics underlying consistent behavior," *Nature neuroscience* **23**(2), 260–270 (2020).

[7] Safaie, M., Chang, J. C., Park, J., Miller, L. E., Dudman, J. T., Perich, M. G., and Gallego, J. A., "Preserved neural dynamics across animals performing similar behaviour," *Nature* **623**(7988), 765–771 (2023).

[8] Degenhart, A. D., Bishop, W. E., Oby, E. R., Tyler-Kabara, E. C., Chase, S. M., Batista, A. P., and Yu, B. M., "Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity," *Nature biomedical engineering* **4**(7), 672–685 (2020).

[9] Schneider, S., Lee, J. H., and Mathis, M. W., "Learnable latent embeddings for joint behavioural and neural analysis," *Nature* **617**(7960), 360–368 (2023).

[10] Ma, X., Rizzoglio, F., Bodkin, K. L., Perreault, E., Miller, L. E., and Kennedy, A., "Using adversarial networks to extend brain computer interface decoding accuracy over time," *elife* **12**, e84296 (2023).

[11] Karpowicz, B. M., Ali, Y. H., Wimalasena, L. N., Sedler, A. R., Keshtkaran, M. R., Bodkin, K., Ma, X., Rubin, D. B., Williams, Z. M., Cash, S. S., et al., "Stabilizing brain-computer interfaces through alignment of latent dynamics," *Nature Communications* **16**(1), 4662 (2025).

[12] Gosztolai, A., Peach, R. L., Arnaudon, A., Barahona, M., and Vandergheynst, P., "Marble: interpretable representations of neural population dynamics using geometric deep learning," *Nature Methods* , 1–9 (2025).