Neural Latent Aligner: Cross-trial Alignment for Learning Representations of Complex, Naturalistic Neural Data

Cheol Jun Cho [1] Edward F. Chang [2] Gopala K. Anumanchipalli [1 2]

Abstract

Understanding the neural implementation of complex human behaviors is one of the major goals in neuroscience. To this end, it is crucial to find a true representation of the neural data, which is challenging due to the high complexity of behaviors and the low signal-to-ratio (SNR) of the signals. Here, we propose a novel unsupervised learning framework, Neural Latent Aligner (NLA), to find well-constrained, behaviorally relevant neural representations of complex behaviors. The key idea is to align representations across repeated trials to learn cross-trial consistent information. Furthermore, we propose a novel, fully differentiable time warping model (TWM) to resolve the temporal misalignment of trials. When applied to intracranial electrocorticography (ECoG) of natural speaking, our model learns better representations for decoding behaviors than the baseline models, especially in lower dimensional space. The TWM is empirically validated by measuring behavioral coherence between aligned trials. The proposed framework learns more cross-trial consistent representations than the baselines, and when visualized, the manifold reveals shared neural trajectories across trials.

1. Introduction

Recent advance in neural recording technologies has prompted active development of diverse neuralmodels (e.g., Pandarinath et al. 2018b; Pei et al. 2021; Kostas et al. 2021), and successful applications have provided significant insights into the principles of neural computation and representation (Pandarinath et al., 2018b; Saxena & Cunningham, 2019; Vyas et al., 2020). Furthermore, these models often provide well-constrained representations that are crucial for developing robust brain-computer interfaces (BCIs) (Pandarinath et al., 2018a; Dyer et al., 2017; Karpowicz et al., 2022; Dabagia et al., 2022). However, application of these models to neural data from naturalistic experiments has been limited, despite the rich implications that such data can offer (Kay et al., 2008; Huth et al., 2016; Silbert et al., 2014; Chartier et al., 2018).

A major challenge is that the stimuli and tasks in naturalistic experiments are highly complex, while existing methods have been developed based on conventional experiments with simple task structures. Previous methods assume the low dimensionality of the task structure or simple dynamics (Pandarinath et al., 2018b; Ye & Pandarinath, 2021). Neither assumption is applicable to real-world behaviors. However, regardless of complexity, behaviorally relevant representations should encode consistent information when the corresponding behaviors are the same. We leverage this hypothesis to learn representations that exclusively encode information consistent across repeated trials. Another significant obstacle is the temporal misalignment among trials caused by the varying durations of behaviors, which hinders direct comparisons across trials. When properly aligned, a novel pattern can be unveiled (Williams et al., 2020).

Here, we suggest a new unsupervised learning framework, Neural Latent Aligner (NLA), to align latent representations across trials, in both time domain and the representational space. NLA outputs a temporal factor called *content factor*, designed to exclusively represent cross-trial consistent information without noise. To achieve this, we propose a new contrastive loss based on InfoNCE (Oord et al., 2018) by incorporating the repeated trials as an additional sampling dimension. The objective is elaborated in Methods (§3.1 & 3.2.3). To address the issue of temporal misalignment, we develop a new time warping model (TWM) with a novel parametrization of temporal alignments. Additionally, we propose two variants of our models, NLA-sDTW and NLA-SUP, which employ distinct approaches for solving the temporal misalignment.

[*]Equal contribution [1]Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA [2]Department of Neurological Surgery, University of California, San Francisco, San Francisco, CA, USA. Correspondence to: Cheol Jun Cho <cheoljun@berkeley.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

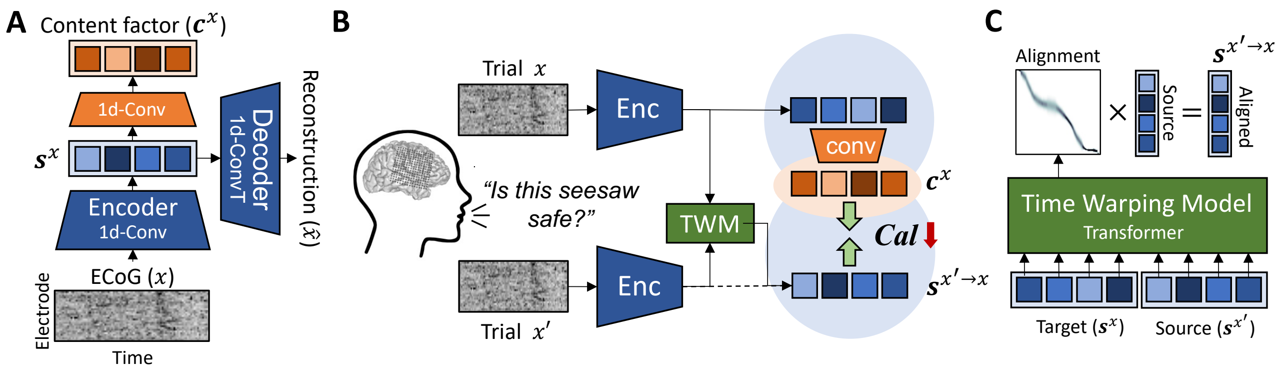


Figure 1. Overview of the proposed framework. **A**) Sequential autoencoder with 1D CNNs for encoding s^x and c^x (content factor). **B**) Diagram of the proposed cross-trial alignment. Two trials of the same behaviors are provided to the pipeline to minimize the Contrastive alignment loss (*Cal*). **C**) Time warping model for modeling alignment distribution from source to target.

We apply our proposed framework to human-speaking neural data, which consists of high-density intracranial electrocorticography (ECoG) measured from two participants while they read full sentences aloud multiple times. Natural speaking involves the complex coordination of multiple motor tasks, making it suitable to demonstrate the effectiveness of our approach and the limitations of previous methods.

We evaluate our method in three aspects: **1)** behavioral relevance of learned representations, **2)** behavioral coherence of unsupervised alignment, and **3)** cross-trial consistency. In all three aspects, our model outperforms the baselines: SeqVAE, LFADS (Pandarinath et al., 2018b), and NDT (Ye & Pandarinath, 2021) (§3.4.1).

Our major contributions are as follows:

- We propose a novel approach, NLA, for aligning neural data across trials to obtain a true representation from noisy neural data.
- We introduce a novel parametrization of temporal alignment that can be implemented as a fully differentiable neural network.
- The content factor of NLA shows the highest correlations with behaviors, in both high- and low-dimensional settings.
- Our TWM can align trials more coherently than other unsupervised alignment methods.
- The content factor exhibits greater consistency across trials compared to baseline representations.
- The manifold of the content factor reveals shared neural trajectories and dynamical structures across trials.

2. Related Work

2.1. Representation learning of neural data

Variational autoencoders (VAEs) (Kingma & Welling, 2013), and its variants have been widely employed for modeling neural data (Zhou & Wei, 2020; Khemakhem et al., 2020; Pandarinath et al., 2018b; Keshtkaran et al., 2022; Liu et al., 2021). In particular, sequential VAEs with dynamic priors have been successful in recovering latent dynamical systems of neural processes (Pandarinath et al., 2018b; Keshtkaran et al., 2022). Some studies try to learn neural representations invariant to random perturbation (e.g., dropping or time-jittering) (Azabou et al., 2021; Liu et al., 2021). Inspired by BERT, predicting randomly masked parts of neural spiking can effectively learn representations of motor neurons (Devlin et al., 2018; Ye & Pandarinath, 2021). Contrastive learning has been applied to large-scale EEG data (Kostas et al., 2021; Défossez et al., 2022). However, learning good representations from noisy neural data becomes challenging when the amount of available data is insufficient compared to the task dimensionality.

2.2. Contrastive learning

Contrastive learning aims to learn representations that are invariant to variance within positive samples but distinctive from negative samples. This approach has been extensively utilized in the field of self-supervised learning (Henaff, 2020; Chen et al., 2020; Baevski et al., 2020; Radford et al., 2021). The most relevant work to our study is SimCLR (Chen et al., 2020), which introduces the concept of generating noisy versions of the data through data augmentation. Here, we obtain the noisy instances from pool of repeated trials rather than from data augmentation.

2.3. Aligning temporally misaligned representations

Dynamic Time Warping (DTW) (Rabiner & Juang, 1993; gio) is a technique used to find an optimal warping path with minimum frame-wise distance between two sequences. Soft-DTW (Cuturi & Blondel, 2017) is a differentiable version of DTW that allows for backpropagation through the visited paths with weighting, not only the optimal path.

DTW has been applied in representation learning to find features invariant to different durations (Haresh et al., 2021; Khaertdinov & Asteriadis, 2022). Previous works emphasize the importance of incorporating contrast in the loss function to prevent representation degeneration.

Williams and colleagues propose to use TWM to align neural signals and reveal previously unseen patterns (Williams et al., 2020). However, their TWM is constrained to be piecewise linear to avoid overfitting caused by high levels of noise, and it requires a specific template for each behavior, limiting its generalizability to unseen behaviors. These limitations make their TWM unsuitable for complex and diverse behaviors.

3. Methods

3.1. Problem formulation

Given a behavior ($y \in \mathcal{Y}$), a set of neural representations is defined as $\mathcal{D}_y$, where each data point ($x \in \mathcal{D}_y$) is an instance of noisy representations of y (e.g., raw neural signals). Our objective is to find a mapping ($\hat{f}$) that maximizes the mutual information (MI, I) between the mapped representation ($f(x)$) and the true representation of y (denoted as x^*) (1).[1]

$$\hat{f} = \underset{f}{\operatorname{argmax}}\ I(f(X); X^*) \tag{1}$$

The above MI maximization can be approximated by minimizing the contrastive loss function (2) (Oord et al., 2018).

$$\mathop{\mathbb{E}}_{y\in\mathcal{Y}, x\in\mathcal{D}_y}\left[-\log\frac{\text{sim}(f(x), x^*)}{\sum_{\tilde{x}\in\mathcal{D}^*_{\text{neg}}}\text{sim}(f(x),\tilde{x})}\right] \tag{2}$$

This constrastive loss function encourages the representation mapped from noisy signals to be closer to the true signal while being distinctive from negative samples ($\tilde{x} \in \mathcal{D}^*_{\text{neg}}$).

However, since x^* is not accessible, the loss function is further approximated by additional sampling of x' (3).

$$\mathop{\mathbb{E}}_{y\in\mathcal{Y}}\left[\mathop{\mathbb{E}}_{(x,x')\in\mathcal{D}_y}\left[-\log\frac{\text{sim}(f(x), x')}{\sum_{\tilde{x}\in\mathcal{D}_{\text{neg}}}\text{sim}(f(x),\tilde{x})}\right]\right] \tag{3}$$

Now, a pair is sampled from $\mathcal{D}_y$, which requires two dimensions of the sampling pool: 1) a representative set of diverse behaviors ($\mathcal{Y}$), and 2) multiple repetitions of each behavior ($\mathcal{D}_y$). The datasets described in §3.3.1 satisfy these conditions.

[1]Capital letters mean corresponding random variables.

3.2. Neural Latent Aligner

3.2.1. Sequential autoencoder

A sequential autoencoder is adopted to model a non-linear projection to latent space. This enables measuring similarities in the latent space of signals, thereby shaping the manifold of learned representations to fulfill the objective. Both the encoder (Enc) and decoder (Dec) are designed as stacks of three 1D convolutional layers with residual connections (more details can be found in Appendix.A). The mapping function f is simply implemented by filtering the outputs of the encoder using two layers of 1D convolution. The autoencoder backbone is trained to minimize the mean squared error of reconstruction ($\mathcal{L}_{\text{recon}} := \mathbb{E}_x\big[(x - \text{Dec}(\text{Enc}(x)))^2\big]$).

The content factor, denoted as c^x, is a temporal factor extracted from neural data that captures the representation of our interest ($c^x := f(x)$). On the other hand, s^x represents the auto-encoded factor, obtained through the encoding process ($s^x := \text{Enc}(x)$). It serves as a surrogate of x by having all the information from x including noise. Both factors lie on a d-dimensional space and form a sequence to represent time series of neural activity ($c^x_t, s^x_t \in \mathbb{R}^d$). Here, d is chosen to be sufficiently large (i.e., $d = 256$) to ensure that s^x can effectively capture the full information of the signals.

3.2.2. Time warping model

Considering the alignment of two time series, z^A and z^B, the probability that the j-th time-point of z^B is aligned to the i-th time point of z^A is denoted as $P(i = j|z^A, z^B)$. The probability distribution over j should be **1) unimodal** to be one-to-one mapping, and **2) monotonic** to avoid time-reversing alignment. To address these, we introduce a novel parametrization that satisfies both requirements.

1) Unimodality: The distribution is parametrized as a Gaussian distribution centered at μ_i with a variance of σ_i^2.

2) Monotonicity: The means of the distribution are monotonically increasing by estimating non-negative ”jumps” between adjacent means ($\Delta\mu_i := \mu_i - \mu_{i-1}, \Delta\mu_i \geq 0$).

$$\text{TWM}(z^A, z^B) = [u^{\Delta\mu}, u^{\sigma}] \tag{4}$$

$$\Delta\mu_i = (L_B - \mu_{i-1})\,\sigma(u_i^{\Delta\mu}) \tag{5}$$

$$\mu_i = \sum_{k=0}^{i}\Delta\mu_k \tag{6}$$

$$\sigma_i^2 = \exp(u_i^{\sigma}) \tag{7}$$

The alignment variables, $[u^{\Delta\mu}, u^{\sigma}]$, are defined for aligning

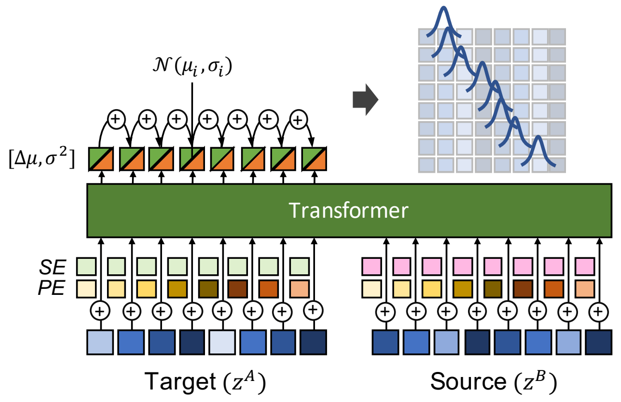


Figure 2. Architecture of the proposed neural time warping model.

the source (z^B), to the target (z^A)(4), with the length of L_B and L_A, respectively. The first variable, $u^{\Delta\mu}$, is passed through the sigmoid function ($\sigma(\cdot)$) and multiplied by the length of the remaining sequence ($L_B - \mu_{i-1}$) (5). Starting from an initial zero mean ($\mu_0 = 0$), the mean at each time point is obtained by accumulating jumps up to that point (6). This formulation allows the means to increase monotonically while staying in the range of the source. Lastly, u^σ is the logarithm of the variance (7).

$$P(i = j|z^A, z^B) \propto \exp\left(\frac{-(j-\mu_i)^2}{2\sigma_i^2}\right) \tag{8}$$

$$z_i^{B\to A} = \sum_{j=1}^{L_B} P(i = j|z^A, z^B)\, z_j^B \tag{9}$$

The final distribution is parameterized as a Gaussian distribution (8), and it is divided by the normalization factor, $\sum_{k=1}^{L_B} \exp(\frac{-(k-\mu_i)^2}{2\sigma_i^2})$. The final warped sequence is then obtained by the expectation (or weighted sum) (9).

This parametrization enables a neural network implementation of TWM$(\cdot)$ and for this particular application, we utilize the Transformer architecture (Vaswani et al., 2017). The multi-head attention mechanism in Transformer is highly effective in incorporating comparisons within and across sequences. The model takes the target and source sequences as input, and at each time point, two learnable encodings are added: positional encoding to encode the position in the sequence (PE$_{i,1\leq i\leq L}$), and sequence encoding to distinguish between the two sequences (SE$_{i,i\in\{1,2\}}$). An overview of the model is depicted in Figure 2.

3.2.3. Contrastive alignment loss

As x and x' are sequential representations, the similarity is factored by each time point, and summed over the sequence in the log space (10). The exponential of the inner product is used for measuring the similarity.[2]

$$\mathop{\mathbb{E}}_{y\in\mathcal{Y}}\left[\mathop{\mathbb{E}}_{(x,x')\in\mathcal{D}_y}\left[-\sum_t^{L_x}\log\frac{\text{sim}(c_t^x, s_t^{x'\to x})}{\sum_k^{L_x}\text{sim}(c_k^x, s_t^{x'\to x})}\right]\right] \tag{10}$$

The negative samples are also adjusted to be pooled along the sequence of content factors. This modification is crucial for stabilizing the training process and encouraging accurate signal warping.[3] We refer to this function as Contrastive alignment loss *(Cal)*.

3.2.4. Final loss function

The final training objective is a weighted sum of the reconstruction loss, the contrastive alignment loss, and additional L2 loss on the content factor ($\mathcal{L}_{\text{L2}}^c$) (11). We set $\lambda_1 = 0.1, \lambda_2 = 0.001$, after some explorations.

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_1\, Cal + \lambda_2\, \mathcal{L}_{\text{L2}}^c \tag{11}$$

3.3. Dataset

3.3.1. Dataset description

We apply our framework to ECoG collected by Anumanchipalli et al. (2019), where participants read aloud full sentences from MOCHA-TIMIT (Wrench, 1999). The MOCHA-TIMIT dataset covers a wide range of natural articulations, thus satisfying the first condition of Equation 3. Our framework also requires multiple trials of the same behavior to perform the two-fold sampling, sampling a behavior and sampling noisy instances of the behavior. For this purpose, we selected two specific participants, S1 and S2, from the original dataset. S1 spoke 50 sentences for 9.40 ± 0.63 times, and S2 spoke 450 sentences for 2.53 ± 0.50 times. These sentences are randomly sampled from MOCHA-TIMIT. The duration for each trial is variable to be 2.29 ± 0.53 seconds for S1, and 2.93 ± 0.81 seconds for S2. We split the dataset based on sentences. The total durations for [train, valid, test] sets are [943s, 42s, 93s] for S1, and [2940s, 70s, 327s] for S2.[4] No trial of the test sentences is included in the train or valid set.

The high-density 16×16 grids were used to collect the ECoG data from brain regions involved in speech: the ventrals sensorimotor cortex (vSMC), superior temporal gyrus (STG), and inferior frontal gyrus (IFG). The signals are processed to high-gamma amplitudes and downsampled to 200 Hz (Chartier et al., 2018; Anumanchipalli et al., 2019). All

[2]The inner product demonstrates more stable training than other metrics. For more details, please refer to Appendix E.1.

[3]Contrasting over warped factors makes training unstable.

[4]The number of sentences in each split [train, valid, test] is [45, 5, 5] for S1 and [400, 10, 40] for S2. For S1, 5 sentences in the valid set are also in the train set but the trials are not overlapped.

256 channels are included and the signals are buffered by 1 second before and after each speaking event.

For the behavioral labels, articulatory kinematic trajectories (AKTs) are extracted by an audio-to-articulatory inversion (AAI) model that is trained on electromagnetic articulography (EMA) corpora (Wrench, 1999; Richmond et al., 2011). Each time point is labeled with 12-dimensional articulatory representations, X,Y coordinates of 6 articulators (Figure 3 A). In addition to AKT, we use the frame-wise phoneme labels to evaluate the models. See Appendix B for detail.

3.3.2. Online alignment of training dataset

For each iteration, a batch of sentences is sampled, and then a pair of repeated trials are sampled for each sentence. The pair is not allowed to be identical, since the pool of possible pairs is not large (especially for S2). Then, the 2 seconds of the signals are sampled for each trial of the pair. As this window size does not encompass the entire sentence, we roughly align the trials prior to sampling to ensure the sampled pairs have overlapping content to be aligned.

Assuming we have a roughly estimated alignment between two trials, the onset of the window is randomly sampled from one of the pair, and then, this onset is warped to the other trial to anchor the window of the counterpart.[5] We update the alignment of training data for every 10K of iterations by applying DTW on the content factor. The initial alignment is set as identity function. This online alignment in the middle of training would not be accurate, but it is sufficient to guide the training.

3.4. Baselines

3.4.1. Baseline representations

We compare our approach to the following baselines:

- **SeqVAE**: A variational version of our model that minimizes ELBO (Kingma & Welling, 2013), instead of the cross-trial alignment. Following previous works (Li & Mandt, 2018; Zhu et al., 2020; Lian et al., 2022), we use a learnable dynamic prior modeled by LSTM.
- **LFADS**: A sequential VAE based on dynamical system, which is designed to find latent factors of the neural process to infer neural population dynamics (Pandarinath et al., 2018b; Keshtkaran et al., 2022).
- **NDT**: Neural Data Transformer (NDT) is a masked autoencoder for neural data (Ye & Pandarinath, 2021). The model is composed of Transformers that learns representations by predicting randomly masked inputs.

The rationale for this selection is that **1)** SeqVAE shares the backbone autoencoder with NLA, and thus, provides direct comparisons with NLA, **2)** LFADS has been frequently employed in this field, and **3)** NDT is chosen since the masked autoencoder has been proposed for a de facto standard of representation learning in many domains (Devlin et al., 2018; He et al., 2022; Tamkin et al., 2022).

For VAEs, the weight (β) for the KL-divergence term is searched over $[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$.[6] We also train LFADS with two different sizes of the latent factor, 32 and 256. NDT is implemented with six layers of Transformer, and the representation of each layer is considered separately. The best layer is chosen by evaluation on the validation data. All the resulting representations have a size of 256 if not specified. See Appendix C for the implementation details.

3.4.2. NLA variants

To show the effectiveness of the cross alignment loss in general, we suggest two variants of our framework. These variants share the objective but differ in the temporal alignment method, replacing the suggested TWM.

- **NLA-sDTW**: Instead of TWM, optimal time warping paths are found by DTW. Training with deterministic alignment from DTW is unstable, thus we use soft-DTW (Cuturi & Blondel, 2017) to make the alignment differentiable. However, soft-DTW calculates the loss by tracking down DTW algorithm, and does not directly provide the alignment distribution nor output the warped series. Therefore, we use Equation 12 for the distance function to approximate Equation 10.

$$\text{sdtw\_dist}(c_i^x, s_j^{x'}) := -\log \frac{\text{sim}(c_i^x, s_j^{x'})}{\sum_k^{L_x} \text{sim}(c_k^x, s_j^{x'})} \quad (12)$$

- **NLA-SUP**: Instead of finding alignment in unsupervised way, NLA-SUP employs a supervised alignment, which is obtained by applying DTW to AKT labels.

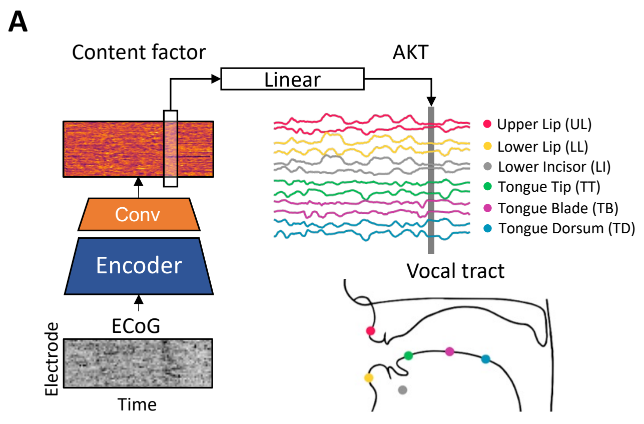
4. Results

4.1. Behavioral relevance of learned representations

The representations from the trained models are evaluated by the performance in linear decoding of behavior (Pei et al., 2021). Here, we use the articulation (AKT) as the reference behavior. AKT is the most prominently represented feature in the brain while speaking (Chartier et al., 2018; Anumanchipalli et al., 2019), and linear decoding of AKT is a robust and valid task for evaluating speech representations (Cho et al., 2023). Together, AKT is an optimal target to inspect the behavioral relevance of the neural representation of the speech. We fit ridge regression to predict AKT from 0.4-second window of content factors, and the correlation

[5]Random jitter of ± 50 ms is added.

[6]Additional $\beta = 1$ is tried for LFADS.



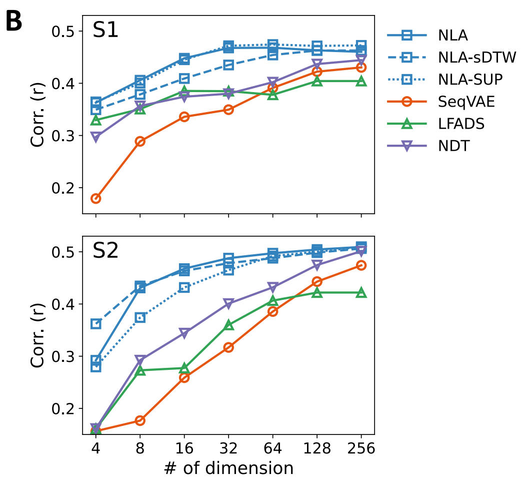


Figure 3. **A**) Overview of behavioral decoding analysis. A linear model is fitted to predict each frame of AKT, for each of XY coordinates of six articulators. **B**) Behavioral decoding performance by reducing dimensionality (blue: NLA, blue-dashed: NLA-sDTW, blue-dotted: NLA-SUP, orange: SeqVAE, green: LFADS, and purple: NDT) The Y-axis means the decoding performance measured as averaged correlation coefficient (r), and the X-axis means the number of dimensionalities. (top: S1, bottom: S2)

Table 1. Performance (r) of linear AKT decoding with full dimensionality (d=256) and reduced dimensionality (d=32).

Model	S1		S2	
	d=32	d=256	d=32	d=256
SeqVAE	0.349	0.430	0.318	0.474
LFADS	0.385	0.404	0.360	0.422
NDT	0.396	0.444	0.401	0.501
NLA	**0.468**	0.460	**0.488**	**0.509**
NLA-sDTW	0.435	**0.463**	0.478	0.506
NLA-SUP	**0.472**	**0.473**	0.465	**0.510**

coefficients on the unseen test sentences are averaged over articulators (Figure 3 A). We also measure this score on the low-dimensional space of the learned representations, where the dimensionality is reduced by PCA.[7] The results are reported in Table 1.

As shown in Table 1, NLA and its variants show higher correlations with AKT than the baselines, for both S1 and S2. The difference from the baselines gets even larger in lower dimensional settings. When we measure AKT correlations by dimensionality, our NLAs are resistant to dimensionality reduction, while the performance of the baselines sharply declines (Figure 3 B).

In general, the performance variance within NLAs is lower than the difference from the baselines. Among NLAs, NLA shows the minimum loss or even gain after the dimensionality is reduced to 32. These results suggest that the cross-trial alignment learns to represent behaviors in a more compact and efficient way, and NLA can learn better content factors than NLA-sDTW, which are comparable to NLA-SUP.

[7]For LFADS, the models trained with explicitly reduced dimensionality are reported if they are better than PCA. See Appendix E.3 for further discussion.

4.2. Empirical validation of time warping model

As a proxy for evaluating the alignment inference, we measure the behavioral coherence between aligned trials. We create additional validation and test datasets by sampling 2-second windows of source-target pairs (valid: 200 pairs, test: 1000 pairs) for each participant. To ensure that the sampled pairs have overlapping contents, we use the supervised alignment of trials to anchor the sampling. [8] Then, we apply unsupervised warping to align the source to the target and check how the phonemes and AKTs are accurately aligned.

When a phoneme in the source is warped, the distance from the reference phoneme in the target trial is measured. Then, the alignment is considered to be correct if the distance is below a threshold. As Figure 4 D, a curve of true positive rate (TPR) can be plotted by adjusting the threshold value, and the area under the curve (AUC:= area/threshold) indicates the sensitivity of the alignment. For AKT, we simply measure the average correlation between the target AKT and the warped source AKT. We additionally report the score by the supervised alignment as a reference for the upper bound of the suggested metrics.

[8]The same sampling method as §3.3.2.

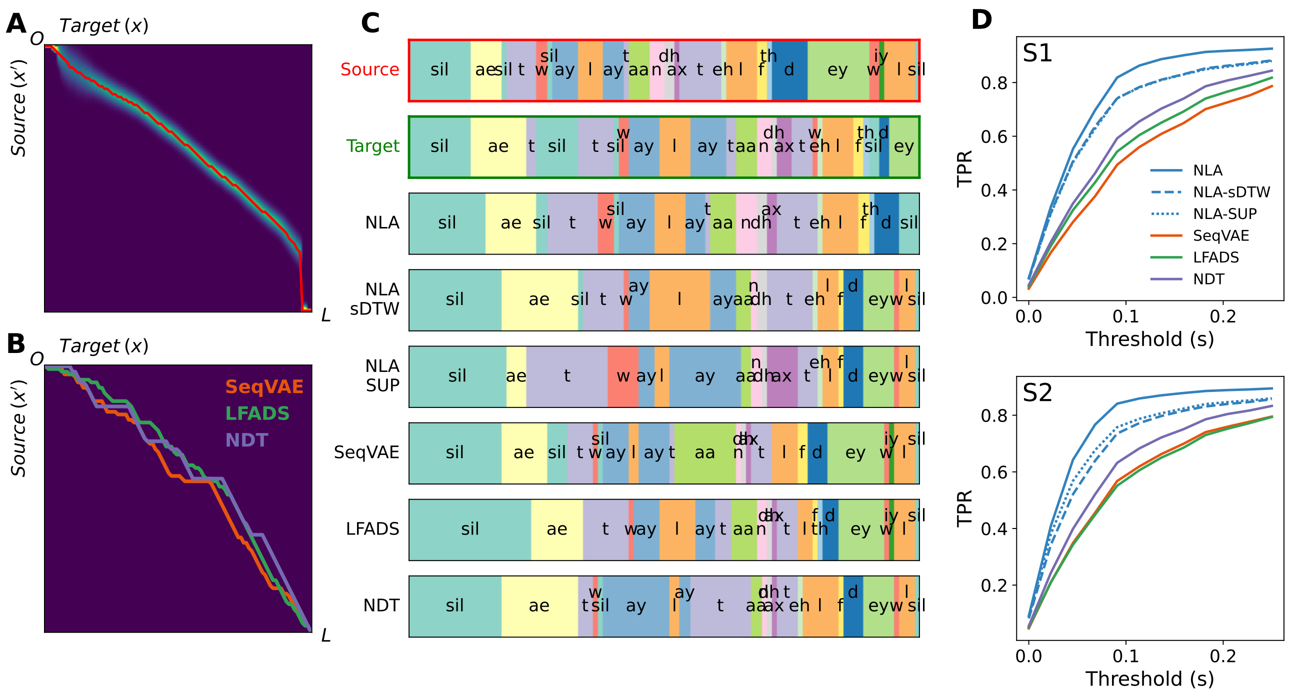


Figure 4. **A)** Example distribution of alignment inferred from TWM. The red line is the warping path estimated by taking the maximum. **B)** Alignments from DTW applied to SeqVAE, LFADS, and NDT, on the same source-target pair as A). **C)** Frame-wise phoneme labels of source (row 1), target (row 2), and source warped by the unsupervised alignment methods (row 3-8). The phoneme is annotated for each color-coded segment. The accuracy of the alignment can be qualitatively measured by inspecting vertical alignment of the segments. **D)** TPR (Y-axis) by time threshold (X-axis) for each subject (top: S1, bottom: S2). The color codes are the same as Figure 3.

Table 2. Aligned phoneme AUC of alignment methods.

Model	S1		S2	
	$\leq 0.5s$	$\leq 2s$	$\leq 0.5s$	$\leq 2s$
SeqVAE	0.683	0.874	0.702	0.868
LFADS	0.711	0.882	0.697	0.867
NDT	0.734	0.888	0.737	0.877
NLA	**0.834**	**0.927**	**0.825**	**0.908**
NLA-sDTW	0.793	0.904	0.777	0.889
NLA-SUP	0.789	0.900	0.785	0.889
Supervised	0.867	0.932	0.849	0.913

Table 3. Aligned AKT correlation of alignment methods.

Model	S1	S2
SeqVAE	0.469	0.495
LFADS	0.483	0.465
NDT	0.555	0.531
NLA	**0.708**	**0.715**
NLA-sDTW	0.667	0.634
NLA-SUP	0.632	0.657
Supervised	0.874	0.851

For our proposed TWM, the alignment is inferred by taking the centers of the output distributions (red line in Figure 4 A). For other models, the alignment is obtained by applying DTW on the representations.[9] The alignment inferred by TWM is smooth (Figure 4 A) but those of other models show a staircase pattern (Figure 4 B). When visually inspected in Figure 4 C, the source phonemes aligned by NLA (the 3rd row) show the highest coherence with the target phonemes (the 2nd row). The threshold-TPR curves of NLA are generally above those of other models, and NLA shows the highest AUC for both S1 and S2 (Table 4.2). The correlations of the aligned AKT are also the highest in NLA (Table 3). This suggests that TWM provides the most accurate unsupervised alignment of behaviors among the methods compared.

[9]See Appendix E.2 for selecting distance function for DTW.

4.3. Cross-trial consistency

We evaluate the cross-trial consistency of the representations to check how well they are aligned within the same behaviors. To make the metric comparable across models, we

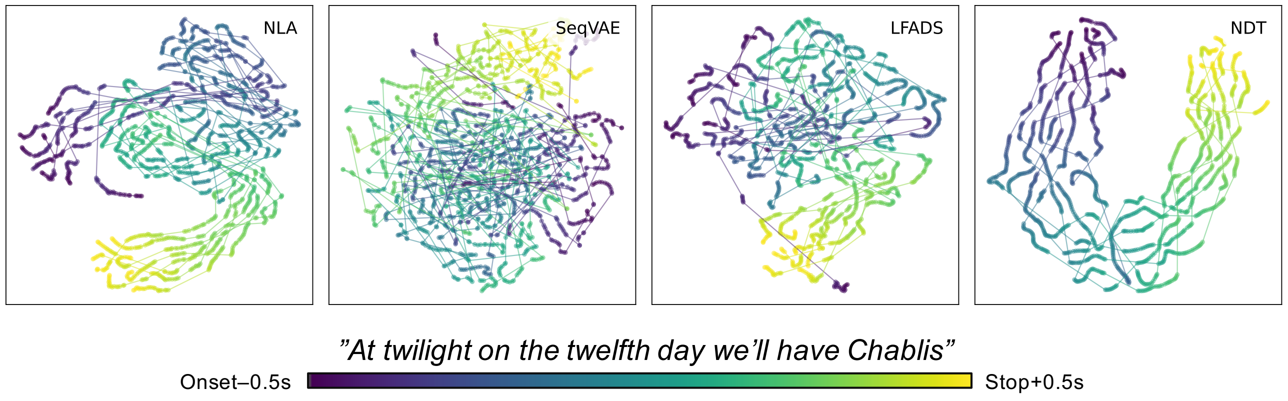


Figure 5. Manifolds visualized by t-SNE (from left to right: NLA, SeqVAE, LFADS, and NDT). The longest sentence with eight repetitions in the test set of S1 is selected: "At twilight on the twelfth day we'll have Chablis." The progress from onset-0.5s to stop+0.5s is colored for each trial individually, and time points in the trial are connected with line. For each model class, the representation with the highest CTC is selected.

Table 4. Cross-trial consistency (CTC) of learned representations.

Model	S1	S2
SeqVAE	0.053	0.046 (0.053)
LFADS	0.121 (0.216)	0.199 (**0.265**)
NDT	0.091 (0.178)	0.182 (0.211)
NLA	**0.281**	**0.230**
NLA-sDTW	0.227	0.186
NLA-SUP	0.273	**0.242**

measure weighted correlations in the PCA space. The correlation of the target (z^A) and the source (z^B) is measured on each PC (e.g., the i-th PC) as $\rho_i = \text{corr}(z^A v_i, z^B v_i)$, and then weighted by the variance of the data explained by the PC (w_i). Finally, the cross-trial consistency (CTC) is defined as summation of the weighted PC correlations ($\text{CTC}(Z) := \sum_i^d w_i \rho_i$). The resulting score lies in [-1, 1]. Here, we use the supervised alignment (i.e., DTW on AKT) for all models, to make the evaluation fair. The CTC for each model class is reported on Table 4. Since there would be a trade-off between representation capacity and cross-trial consistency, we select hyperparameters that yield the highest decoding performance or the highest CTC. If different, we report the latter in parentheses (Table 4).

For S1, NLA shows the highest CTC, which is even higher than the supervised version (NLA-SUP). For S2, the CTC score of NLA is also close to that of NLA-SUP, and higher than the baselines except for LFADS with 0.265. This particular LFADS instance has a decoding performance of 0.327, which is significantly low based on Table 3. Compared to NLA-sDTW, NLA in both participants outperforms NLA-sDTW by a large margin. These results demonstrate that our NLA can learn representations consistent across trials.

When the manifold is visualized by t-SNE, shared neural trajectories and dynamics are easily identifiable in NLA (Figure 5 leftmost panel). NDT also shows a general trend (Figure 5 rightmost panel) but with coarse dynamics compared to NLA. The representations near the stop are clustered in SeqVAE (Figure 5 second left panel) but no visible cluster is found in other time points. LFADS show some shared dynamics but not as clear as NLA (Figure 5 second right panel). More examples can be found in Appendix D.2.

5. Discussion

The content factor extracted by our model encodes behaviors with compact dimensionality. This property has the potential to enhance the sample efficiency and robustness of BCI applications, which is crucial given the expense and limited nature of data collection, especially in invasive recording.

The definition of consistency used in our framework is flexible, so that it can be defined to be across subject to investigate neurotypical principles, or across session to stabilize long term deployment of BCI (Dyer et al., 2017; Karpowicz et al., 2022). Tasks in other domain, such as action recognition, can be revisited with our framework. Exploring these broader applications is our future direction.

Our method reveals a more interpretable and meaningful manifold than the previous methods, although the high-density ECoG is a challenging modality. Therefore, the method can be utilized to investigate the neural manifolds and dynamics of complex behaviors, without relying on external labels. As far as we know, this is the first demonstration of the unsupervised discovery of a clean manifold from highly noisy and complex neural data.

Limitation: Our approach requires multiple repetitions for the same condition, which may not be applicable to some experiments (e.g., spontaneous speech).

6. Conclusion

We propose a novel unsupervised learning framework to learn representation from noisy neural data by cross-trial alignment. To this end, we develop a contrastive alignment loss and a differentiable time warping model. The effectiveness of our proposed framework is empirically demonstrated with challenging human ECoG data.

7. Acknowledgements

This research is supported by the following grants to PI Anumanchipalli — NSF award 2106928, Rose Hills Foundation and Noyce Foundation.

References

Anumanchipalli, G. K., Chartier, J., and Chang, E. F. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498, 2019.

Azabou, M., Azar, M. G., Liu, R., Lin, C.-H., Johnson, E. C., Bhaskaran-Nair, K., Dabagia, M., Avila-Pires, B., Kitchell, L., Hengen, K. B., et al. Mine your own view: Self-supervised learning through across-sample prediction. *arXiv preprint arXiv:2102.10106*, 2021.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

Chartier, J., Anumanchipalli, G. K., Johnson, K., and Chang, E. F. Encoding of articulatory kinematic trajectories in human speech sensorimotor cortex. *Neuron*, 98(5):1042–1054, 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Internationals conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Cho, C. J., Wu, P., Mohamed, A., and Anumanchipalli, G. K. Evidence of vocal tract articulation in self-supervised learning of speech. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Cuturi, M. and Blondel, M. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pp. 894–903. PMLR, 2017.

Dabagia, M., Kording, K. P., and Dyer, E. L. Comparing high-dimensional neural recordings by aligning their low-dimensional latent representations. *arXiv preprint arXiv:2205.08413*, 2022.

Défossez, A., Caucheteux, C., Rapin, J., Kabeli, O., and King, J.-R. Decoding speech from non-invasive brain recordings. *arXiv preprint arXiv:2208.12266*, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dyer, E. L., Gheshlaghi Azar, M., Perich, M. G., Fernandes, H. L., Naufel, S., Miller, L. E., and Körding, K. P. A cryptography-based approach for movement decoding. *Nature biomedical engineering*, 1(12):967–976, 2017.

Giorgino, T. Computing and visualizing dynamic time warping alignments in r: the dtw package. *Journal of statistical Software*, 31:1–24, 2009.

Haresh, S., Kumar, S., Coskun, H., Syed, S. N., Konin, A., Zia, Z., and Tran, Q.-H. Learning by aligning videos in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5548–5558, 2021.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Henaff, O. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pp. 4182–4192. PMLR, 2020.

Huth, A. G., De Heer, W. A., Griffiths, T. L., Theunissen, F. E., and Gallant, J. L. Natural speech reveals the semantic maps that tile human cerebral cortex. *Nature*, 532 (7600):453–458, 2016.

Karpowicz, B. M., Ali, Y. H., Wimalasena, L. N., Sedler, A. R., Keshtkaran, M. R., Bodkin, K., Ma, X., Miller, L. E., and Pandarinath, C. Stabilizing brain-computer interfaces through alignment of latent dynamics. *bioRxiv*, 2022.

Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. Identifying natural images from human brain activity. *Nature*, 452(7185):352–355, 2008.

Keshtkaran, M. R., Sedler, A. R., Chowdhury, R. H., Tandon, R., Basrai, D., Nguyen, S. L., Sohn, H., Jazayeri, M., Miller, L. E., and Pandarinath, C. A large-scale neural network training framework for generalized estimation of single-trial population dynamics. *Nature Methods*, 19 (12):1572–1577, 2022.

Khaertdinov, B. and Asteriadis, S. Temporal feature alignment in contrastive self-supervised learning for human activity recognition. *arXiv preprint arXiv:2210.03382*, 2022.

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217. PMLR, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kostas, D., Aroca-Ouellette, S., and Rudzicz, F. Bendr: using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data. *Frontiers in Human Neuroscience*, pp. 253, 2021.

Li, Y. and Mandt, S. Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*, 2018.

Lian, J., Zhang, C., and Yu, D. Robust disentangled variational speech representation learning for zero-shot voice conversion. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6572–6576. IEEE, 2022.

Liu, R., Azabou, M., Dabagia, M., Lin, C.-H., Gheshlaghi Azar, M., Hengen, K., Valko, M., and Dyer, E. Drop, swap, and generate: A self-supervised approach for generating neural activity. *Advances in Neural Information Processing Systems*, 34:10587–10599, 2021.

Maghoumi, M., Taranta, E. M., and LaViola, J. Deepnag: Deep non-adversarial gesture generation. In *26th International Conference on Intelligent User Interfaces*, pp. 213–223, 2021.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Pandarinath, C., Ames, K. C., Russo, A. A., Farshchian, A., Miller, L. E., Dyer, E. L., and Kao, J. C. Latent factors and dynamics in motor cortex and their application to brain–machine interfaces. *Journal of Neuroscience*, 38 (44):9390–9401, 2018a.

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, 2018b.

Pei, F., Ye, J., Zoltowski, D., Wu, A., Chowdhury, R. H., Sohn, H., O'Doherty, J. E., Shenoy, K. V., Kaufman, M. T., Churchland, M., et al. Neural latents benchmark'21: Evaluating latent variable models of neural population activity. *arXiv preprint arXiv:2109.04463*, 2021.

Rabiner, L. and Juang, B.-H. *Fundamentals of speech recognition*. Prentice-Hall, Inc., 1993.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Richmond, K., Hoole, P., and King, S. Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

Saxena, S. and Cunningham, J. P. Towards the neural population doctrine. *Current opinion in neurobiology*, 55: 103–111, 2019.

Silbert, L. J., Honey, C. J., Simony, E., Poeppel, D., and Hasson, U. Coupled neural systems underlie the production and comprehension of naturalistic narrative speech. *Proceedings of the National Academy of Sciences*, 111 (43):E4687–E4696, 2014.

Tamkin, A., Banerjee, G., Owda, M., Liu, V., Rammoorthy, S., and Goodman, N. Dabs 2.0: Improved datasets and algorithms for universal self-supervision. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vyas, S., Golub, M. D., Sussillo, D., and Shenoy, K. V. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249, 2020.

Williams, A. H., Poole, B., Maheswaranathan, N., Dhawale, A. K., Fisher, T., Wilson, C. D., Brann, D. H., Trautmann, E. M., Ryu, S., Shusterman, R., et al. Discovering precise temporal patterns in large-scale neural recordings through robust and interpretable time warping. *Neuron*, 105(2): 246–259, 2020.

Wrench, A. Mocha: multichannel articulatory database. http://www.cstr.ed.ac.uk/research/project/artic/mocha.html, 1999.

Ye, J. and Pandarinath, C. Representation learning for neural population activity with neural data transformers. *arXiv preprint arXiv:2108.01210*, 2021.

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

Zhu, Y., Min, M. R., Kadav, A., and Graf, H. P. S3vae: Self-supervised sequential vae for representation disentanglement and data generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6538–6547, 2020.

A. Implementation detail

A.1. Sequential autoencoder

The backbone of our proposed famework is a sequential autoencoder with 1D CNN. Both encoder and decoder are composed of two layers of convolutional layers with residual connection (ResLayer). The configurations are shown in the Table A.1. The parameters are denoted as the PyTorch convention: conv1d (in channel,out channel, kernel, stride, dilation), and convT1d (in channel, out channel, kernel, stride). The mapping, f, is defined as two stacks of conv1d (256, 256, 3, 1, 1) on top of the encoder. The activation function, ReLU, is applied for the output of every conv1d, and followed by Batch normalization. The input to this model is in the shape of (Batch size (B), T=400, number of channels=256), and the resulting latent factors have (B, L=100, d=256).

Table 5. Architecture of the backbone sequential autoencoder.

Encoder	
ResLayer1	conv1d (256, 256, 3, 2, 1)
	conv1d (256, 256, 3, 1, 2)
ResLayer2	conv1d (256, 256, 3, 2, 1)
	conv1d (256, 256, 3, 1, 2)
Linear	fc (256, 256)
Decoder	
Upsample	convT1d (256, 256, 2, 1)
ResLayer1	conv1d (256, 256, 3, 1, 1)
	conv1d (256, 256, 3, 1, 1)
Upsample	convT1d (256, 256, 2, 1)
ResLayer2	conv1d (256, 256, 3, 1, 1)
	conv1d (256, 256, 3, 1, 1)
Logit	fc (256, 256)

A.2. Time warping model

The time warping model is implemented with four layers of Transformer (Vaswani et al., 2017). We follow a conventional practice of implementing Transformer. Each Transformer layer is composed of multi-head attention and feed forward modules. Following the same notation in Vaswani et al. (2017), we use $d_{\text{model}} = 64, d_k = 32, h = 4$ for multi-head attention, and $d_{\text{ff}} = 128$ for the feed forward model. ReLU is used for the activation function of the feedforward model, and Dropout rate 0.2 is applied to the attentions and the output of the first layer in the feedforward model. Additional Layer normalization is applied to the output of each module. The proposed time warping model is then composed of an input layer with fc(256, 64), four stacks of the Transformer blocks, and an output layer with fc(64, 2). The positional embedding (PE) and sequential embedding (SE) are added to the input before the Transformer layers. The input to this model is pairs of sequences, [(B, L, d), (B, L, d)], and then the output is also pairs of parameters for alignment distribution [(B, L, 2), (B, L, 2)]. In the main text, we suggest to warp one trial to another using one half of the outputs, but in practice, we align them bidirectionally by leveraging the other half part of the outputs.

For soft-DTW in NLA-sDTW, we use codes provided by Maghoumi et al. (2021), using default setting ($\gamma = 1.0$) other than the proposed distance function. For inferring alignment after training the models except for NLA, we use DTW implemented by Giorgino (2009) and every DTW uses default setting without window.

A.3. Training configuration

We use one NVIDIA RTX A5000 to train the models, and the batch size of 64 is used. We augment the data by randomly dropping 5-10% of channels of randomly chosen 50% of trials in the batch. Adam optimizer is used with $\beta_1 = 0.9, \beta_2 = 0.999$. The number of iterations is 500K and the learning rate is annealed from 1e-3 to 1e-4 using cosine annealing.

B. Data preprocessing detail

B.1. Signal processing

The ECoG with 16 x 16 grids collected raw local field potentials in 3K Hz. The 60 Hz line noise by DC connector is filtered out with a notch filter. Then, the Hilbert transform is applied to extract the analytic amplitude of the high-gamma frequency (70–200 Hz). Lastly, the signals are downsampled to 200 Hz.

B.2. Articulatory kinematic trajectory (AKT)

The audio-to-articulation inversion (AAI) model is trained with bidirectional LSTM on two electromagnetic articulography (EMA) datasets, MOCHA-TIMIT (Wrench, 1999) and MNGU0 (Richmond et al., 2011). The AAI model is trained to predict 12-dimensional (X,Y coordinates for each of 6 articulators) articulatory representations (AKTs) from acoustic features (MFCC). Then, the AKTs are inferred using recorded participants' audio.

B.3. ECoG grid locations

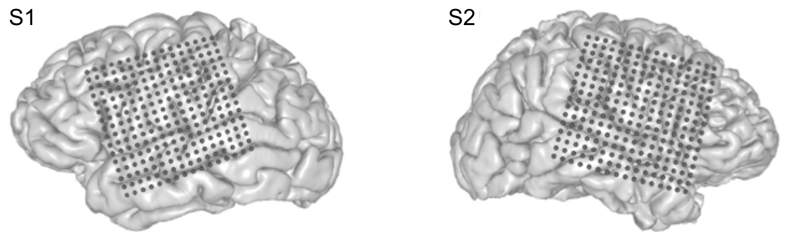


Figure 6. The location of the implanted ECoG (left: S1, right: S2). The figures are from Anumanchipalli et al. (2019)

C. Baseline implementation

C.1. SeqVAE

The variational inference is added to the autoencoder with the same CNN architecture as A.1. A dynamic prior on the latent space is defined in non-parametric way. We follow the ideas and the implementation by the previous deep sequential VAE models (Li & Mandt, 2018; Zhu et al., 2020; Lian et al., 2022).

For each time point, a sample from prior distribution is generated by running LSTM on random vectors sampled from a diagonal Gaussian distribution, $\mathcal{N}(0, I)$. This LSTM is trained along with the VAE and provides sample depend on the sampling history, shaping dynamical prior of the embedding space. We redirect to (Li & Mandt, 2018) for the detailed formulation of the dynamic prior. This LSTM is implemented as a single layer with the hidden size of 256.

We train model for different weighting factors of KL-divergence term in ELBO: [1e-4, 1e-3, 1e-2, 1e-1]. All the training configurations are the same as A.3. The batch size is increased to 128 since the pairs are batched in NLA, so we need to double up the batch size to match the size.

C.2. LFADS

We follow the implementation by Pandarinath et al. (2018b); Keshtkaran et al. (2022), with a bidirectional GRU encoder, and a unidirectional GRU controller and generator. The size of all the hidden states are set as 128, and the size of the factor is set as 32 or 256. For the prior on the initial state, we used a Gaussian distribution with trainable mean and fixed variance of 0.1. The prior on the controller output is autoregressively defined with trainable autocorrelation variables. The input signals are downsampled to 50 Hz to match frequency of the resulting representation from other models. The weight for KL-divergence term is selected from [1e-4, 1e-3, 1e-2, 1e-1, 1], and fixed throughout the training. The number of iterations is set as 50K and the batch size is also doubled. Other settings are identical to A.3.

C.3. NDT

For NDT, a masked autoencoder with the Transformer architecture is implemented. There are some key differences from the original model proposed by Ye & Pandarinath (2021): 1) convolutional positional encoding, 2) randomized masking plan, and 3) modified loss objective to incorporate *unmasked* prediction (autoencoding).

First of all, since Transformer lacks generalizability to unseen lengths, the model trained with the fixed window of inputs fails to model variable length of trials. To tackle this, our NDT takes the full-length trials as inputs, and to do so, we use convolutional positional embedding (Baevski et al., 2020) instead of the lookup table positional embedding. The convolution with kernel size of 128 and group of 8 is applied to the input to extract positional information from the data, and then added to the input before entering the Transformer layers. This convolutional position encoding is proven to be successful in speech domain (Baevski et al., 2020) and agnostic to the input length. Other than that, Transformer has the same architecture as A.2 with $d_{\text{model}} = 256, h = 4, d_k = 64, d_{\text{ff}} = 256$, and six layers are used.

We also diversify the masking probability by randomly selecting from [0.2, 0.4, 0.6]. The input sequence are first segmented and each segment is masked with the chosen masking probability. The length of the segment is randomly chosen from [5, 10]. The model overfits severely if only trained with masked prediction objective. Therefore, we use an auxiliary loss of the reconstruction of the unmasked part. This loss is weighted with 0.1 and added to the masked prediction loss. This modification prevents the model from overfitting. The number of iterations are set as 100K and the other settings are the same as A.3.

D. Additional analyses

D.1. Reconstruction performance

The ECoG signals are highly noisy and there is spontaneous background activity that is not random but behaviorally irrelevant. Thus, we don't compare this in the main body since the reconstruction performance doesn't distinguish noise, failing to be a valid evaluation metric. Table 6 shows the reconstruction performance of each model, measured as average correlation coefficients of 256 channels. We select the model instance by decoding accuracy in validation set and the number in parentheses means the score selected using decoding from low-dimensional models (dim=32) if applicable. The reconstruction performance is maxing out in NLAs and this is natural since the content factor is not directly involved in the reconstructions. So NLA is free of the capacity-regularization trade-off that is shown in the case of SeqVAE and LFADS.

Table 6. Reconstruction accuracy (r)

Model	S1	S2
SeqVAE	0.959 (0.902)	0.963 (0.909)
LFADS	0.721 (0.528)	0.549 (0.765)
NDT	0.907	0.918
NLA	0.992	0.985
NLA-sDTW	0.988	0.984
NLA-SUP	0.990	0.986

D.2. Manifold visualization

The manifolds of other test sentences in S1 are visualized in Figure D.2. The difference across models pointed out in the main text is replicated in other sentences. We also try with higher perplexity (right) and the results show the same conclusion. The dimension of the data is reduced to 32 by PCA prior to running t-SNE.

E. Other design choices

E.1. Distance functions for training NLA

The choice of distance function affects the stability of training the TWM in NLA. The instability of TWM results in degeneration of the inferred alignment (e.g., all the alignment center indices are collapsed to zero or the end). With the Euclidean distance or cosine distance, the training requires more careful hyperparameter selection. However, training TWM

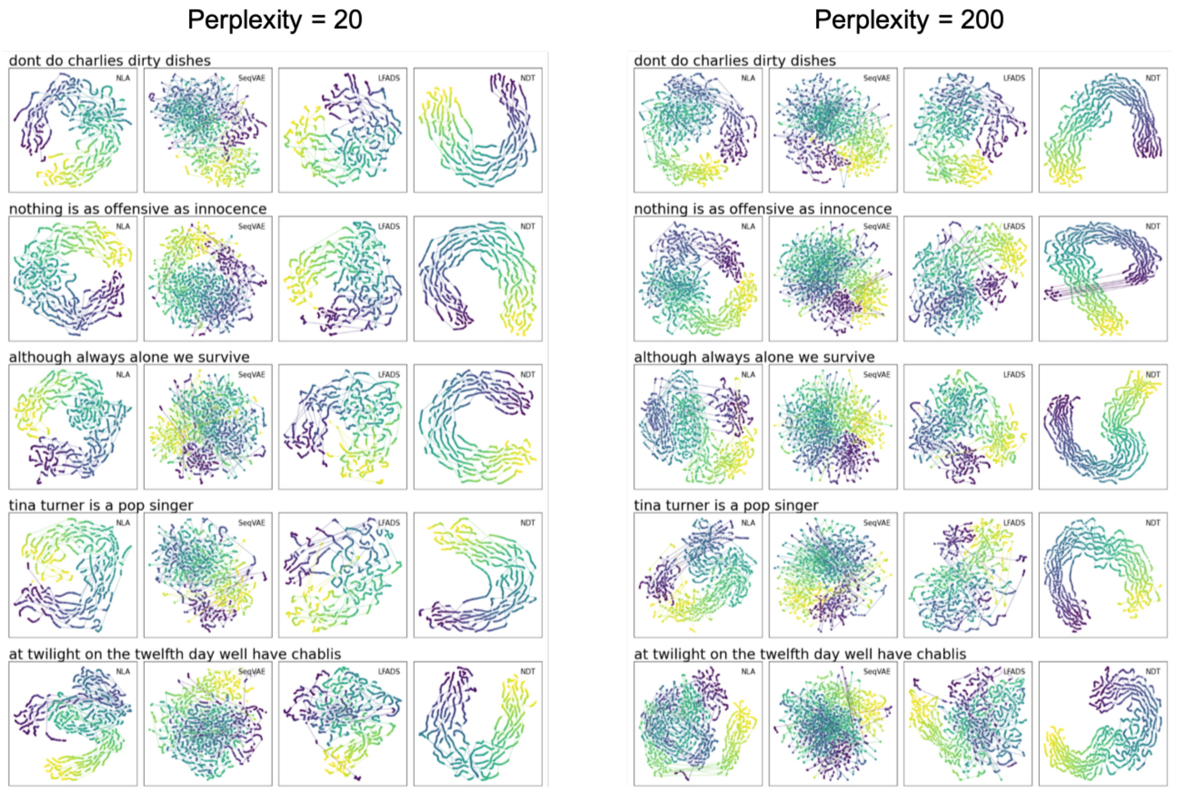


Figure 7. t-SNE visualizations on other test sentences with different perplexities (left: 20, right: 200). The main figure uses perplexity=20.

with the inner product distance was highly stable. Thus, we decided to use the inner product as the similarity function for all experiments. For further stabilization, the inner product values are softly clamped to be in [-5, 5] by $5\tanh(x/5)$. Instead, we experimented with the Euclidean distance and cosine distance for one of our variants, NLA-sDTW. For cosine distance, the similarity is divided by temperature which is set as 0.2. We consistently found lower performance when cosine or the Euclidean distance is used. Table 7 shows the scores of NLA-sDTW with other distance functions: decoding (§4.1), alnPh & alnAKT (§4.2), and CTC (§4.3).

Table 7. Model performance by distance function type

Distance	decoding		alnPh		alnAKT		CTC	
	S1	S2	S1	S2	S1	S2	S1	S2
product	**0.463**	**0.506**	**0.904**	**0.889**	**0.667**	**0.634**	**0.227**	**0.186**
cosine	0.457	0.498	0.895	0.883	0.617	0.596	0.083	0.098
euclidean	0.442	0.503	0.883	0.865	0.596	0.453	0.034	0.042

E.2. Optimal distance function for post-training DTW

As the temporal alignments are not inferred in the baseline models, we use DTW to get the temporal alignments between trials after training the models. We experimented with the Euclidean distance and product distance (negative of inner product) particularly for this post-training DTW. Table 8 shows behavioral coherence scores, alnPh and alnAKT (§4.2), by distance functions for each model. For the baseline models (SeqVae, LFADS, NDT), the product distance is mostly selected as optimal on the validation set. For NLA-sDTW and NLA-SUP, the Euclidean distance is mostly selected as optimal. However, the difference in the scores across distance methods is not as significant as the difference between the scores of NLAs and those of the baselines. The choice of distance has minimal effect in post-training DTW and the alignments inferred by TWM show higher scores than any configurations.

Table 8. Behavioral coherence by distance function type in post-training DTW.

Model	Distance	alnPh		alnAKT	
		S1	S2	S1	S2
SeqVAE	product	**0.874**	**0.868**	**0.469**	**0.495**
	euclidean	0.868	0.862	0.444	0.449
LFADS	product	**0.882**	**0.867**	0.483	**0.465**
	euclidean	0.871	0.859	**0.486**	0.460
NDT	product	0.887	**0.877**	**0.555**	**0.531**
	euclidean	**0.888**	0.873	0.531	0.521
NLA-sDTW	product	0.897	0.889	0.647	0.634
	euclidean	**0.904**	**0.891**	**0.667**	**0.639**
NLA-SUP	product	0.895	0.887	0.632	0.649
	euclidean	**0.900**	**0.889**	**0.664**	**0.657**
NLA	N/A	**0.927**	**0.908**	**0.708**	**0.715**

E.3. Reason for using PCA to reduce the dimensionality of the learned reperesentations

We reduce dimensionality post hoc since models with explicitly reduced dimensionality tend to overfit. The overfitting problem is particularly happening with the reconstruction by the decoder since we only reduced the size of target representations, d, and left other parts the same. Thus, the decoder has the same capacity while the input has lower dimensionality. Therefore, we suspect that overfitting is happening because the decoder is trained to compensate for the neural signals from short input encoding, which is hard to be generalized. This may be mitigated by controlling the size of the decoder as well, but this opens up too much degree of freedom to cover. Thus, we decided to use post hoc dimensionality reduction by PCA due to the limitation of computing resources to explore other possibilities.