**Independent Research Report**

# Brain-to-Brain Decoder
# Validating Neural Synchrony Patterns in Human Conversation

**Author:**

Mohammad Ahmadullah Khan

Department of Computer Engineering, Jamia Millia Islamia, New Delhi
BS in Data Science and Applications, Indian Institute of Technology Madras

2026

## Contents

**Abstract** **2**

**Contributions** **2**

**1 Scientific Rationale and Related Work** **2**

**2 Preliminary Results on a Single Dyad** **4**

**3 Methodology for Extension to Multiple Dyads** **7**

**4 Implementation and System Design** **10**

**5 Goals of the Extended Study** **12**

**6 Planned Stages of Work** **13**

**References** **14**

## Abstract

This study examines how neural activity from two interacting participants can be jointly represented using EEG hyperscanning data. It develops a reproducible pipeline that takes synchronized dual-subject EEG, constructs a joint representation, and learns a low-dimensional embedding using **CEBRA** [3].

The core idea is to treat the dyad as a single system. Preprocessed signals from both participants are concatenated into a shared, time-aligned representation and mapped into a latent space. The embedding is evaluated using simple decoding tasks (k-NN) and goodness-of-fit metrics, together with shuffled-data controls that verify the learned structure is not due to noise or misalignment.

Rather than assuming a specific theory of social interaction, the study asks whether consistent structure can be learned from dyadic data and how stable these representations are under different conditions. On a single dyad, a hybrid CEBRA embedding decodes affective condition with a k-NN accuracy of 0.967, while label-shuffle and temporal-shuffle controls fall to chance. The report also describes how the pipeline extends to multiple dyads, including statistical comparison of embeddings with the cross-entropy test [2], used as a comparison tool rather than a standalone distance metric.

## Contributions

**Reproducible pipeline for dyadic EEG:** A clear and minimal pipeline for synchronizing, preprocessing, and embedding dual-subject EEG data. This reduces ambiguity in handling hyperscanning data and avoids ad-hoc preprocessing choices.

**Validated embedding framework:** Systematic evaluation using controls (shuffling, baseline decoding) checks that learned representations reflect meaningful structure rather than artifacts.

**Extensible setup:** The pipeline is designed to scale to multiple dyads without changing the core logic, making it usable for further experiments on interaction-based neural data.

## 1 Scientific Rationale and Related Work

### 1.1 Communication and the "Mismatch" Hypothesis

Traditional views in autism research often frame communication differences as individual deficits. More recent work shifts this toward an interaction-based perspective, in which communication depends on the pairing between participants rather than on one-sided impairment.

Crompton et al. [1] study this using diffusion chains and show that communication outcomes depend on the interaction context and pairing. However, these results are behavioral, not neural, and do not directly establish how brain activity aligns between participants. This leaves an open question: whether measurable structure exists in joint neural activity during interaction, and whether such structure is consistent across different dyads.

This study does not attempt to validate any specific theory. Instead, it asks whether dyadic neural data contain stable, learnable structure when both participants are modeled together.

### 1.2 Contrastive Representation Learning with CEBRA

CEBRA is a contrastive learning method designed to extract low-dimensional embeddings from neural time-series data [3]. It learns a mapping in which temporally or behaviorally similar samples are placed closer together, while dissimilar samples are pushed apart.

Given an anchor sample $z_i$, a positive sample $z_{pos}$, and a set of negatives, the model optimizes a contrastive objective of the form:

$$\mathcal{L} = -\log \frac{\exp(\text{sim}(z_i, z_{pos})/\tau)}{\sum_j \exp(\text{sim}(z_i, z_j)/\tau)}$$

The learned embedding is not directly interpretable on its own, but it can be evaluated through downstream tasks such as decoding or reconstruction of behavioral variables. A key advantage is that it can use time structure or labels without requiring explicit feature engineering.

### 1.3 Hyperscanning and Dyadic Modeling

Most neural analysis is done on single participants. In contrast, hyperscanning records two or more participants simultaneously during interaction.

This changes the modeling setup. Instead of treating each brain independently, both signals can be combined into a single time-aligned representation. A simple approach is to concatenate channels from both participants, forming a joint matrix of shape $(T \times 128)$ for a 64-channel setup.

Recent work has shown that applying representation learning methods to this joint signal can capture patterns related to interaction roles or shared dynamics. However, these results depend heavily on preprocessing, alignment, and validation choices.

### 1.4 Comparing Embedding Structures

Low-dimensional embeddings are often inspected visually, but visual inspection is unreliable. A more useful direction is to compare embeddings statistically.

The cross-entropy test [2] compares distributions of embeddings by measuring how well one distribution explains another. In this study, it is used to compare embeddings across runs or subsets of data, not as a standalone distance metric.

### 1.5 Positioning of This Work

This study sits at the intersection of:

- dyadic EEG data processing,
- contrastive representation learning,
- and statistical comparison of learned embeddings.

The focus is not on proposing a new theory, but on building a controlled pipeline in which:

- data are properly synchronized,
- embeddings are learned in a consistent way,
- and results are validated with simple, interpretable checks.

## 2 Preliminary Results on a Single Dyad

### 2.1 Dataset Description and Synchronization

The analysis uses a single dyadic EEG hyperscanning recording (two participants, 64 channels each, sampled at 250 Hz), provided by its owners for evaluation purposes; the data are not redistributed with this report. The two recordings were aligned using DIN1 trigger markers.

A constant offset of approximately 217 ms was observed between the two streams. Since this offset remained fixed across the recording and did not drift, it was treated as a startup misalignment rather than a timing instability. The data were cropped based on shared trigger intervals to obtain time-aligned segments.

After alignment:

- Channel 65 (vertex reference) was removed from both participants.
- The two signals were concatenated to form a joint representation of shape ($T \times 128$).
- Each channel was z-normalized independently across time. Within-segment normalization was initially tested but introduced label leakage (forcing segment means to zero), and was therefore replaced with global z-scoring after concatenation.

Table 1: Dataset summary

| **Parameter** | **Value** |
|---|---|
| Channels per participant | 64 |
| Total channels (dyad) | 128 |
| Sampling rate | 250 Hz |
| Alignment method | DIN1 trigger cropping |
| Reference channel | Removed (VREF) |

### 2.2 Preprocessing and Artifact Handling

Preprocessing included bad-channel handling, interpolation, and ICA-based artifact removal. Extended Infomax ICA was applied, and components were rejected using a combination of ICLabel probabilities and visual inspection.

A large fraction of components was removed (22/30 for one participant and 19/30 for the other). This is higher than in typical EEG pipelines and may indicate either strong contamination in the raw signal or over-aggressive rejection. Although post-ICA spectra retained visible alpha and beta activity, the possibility of removing meaningful neural signal cannot be ruled out and needs further validation.

### 2.3 Embedding and Decoding Performance

The joint signal was embedded into a 3D latent space using CEBRA under three configurations: time-based, label-based (affect), and a hybrid combination.

The hybrid configuration achieved the highest performance. The label-based model also shows strong decoding accuracy relative to controls, indicating that the embedding captures structured

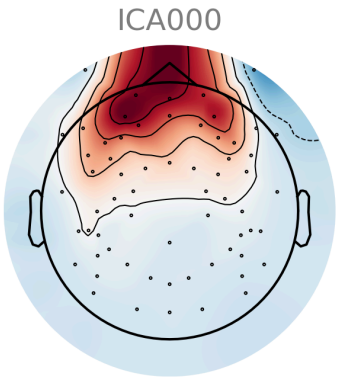



Figure 1: Example ICA component rejected during preprocessing. T000 (listener): frontal-pole bilateral activation, consistent with ocular artifact (EOG).

Table 2: Embedding performance

| **Model** | **k-NN Accuracy** | $R^2$ **(GOF)** |
|---|---|---|
| Time-based | 0.490 | −0.478 |
| Label-based | 0.948 | 0.818 |
| Hybrid | 0.967 | 0.878 |

signal rather than trivial statistics. However, additional runs with different random seeds showed variability in hybrid performance, indicating sensitivity to initialization.

The embedding geometry shows partial separation between labeled segments, with overlap and outliers present. This suggests that the learned representation is not strictly binary but distributed, and may reflect a combination of condition differences and residual variability.

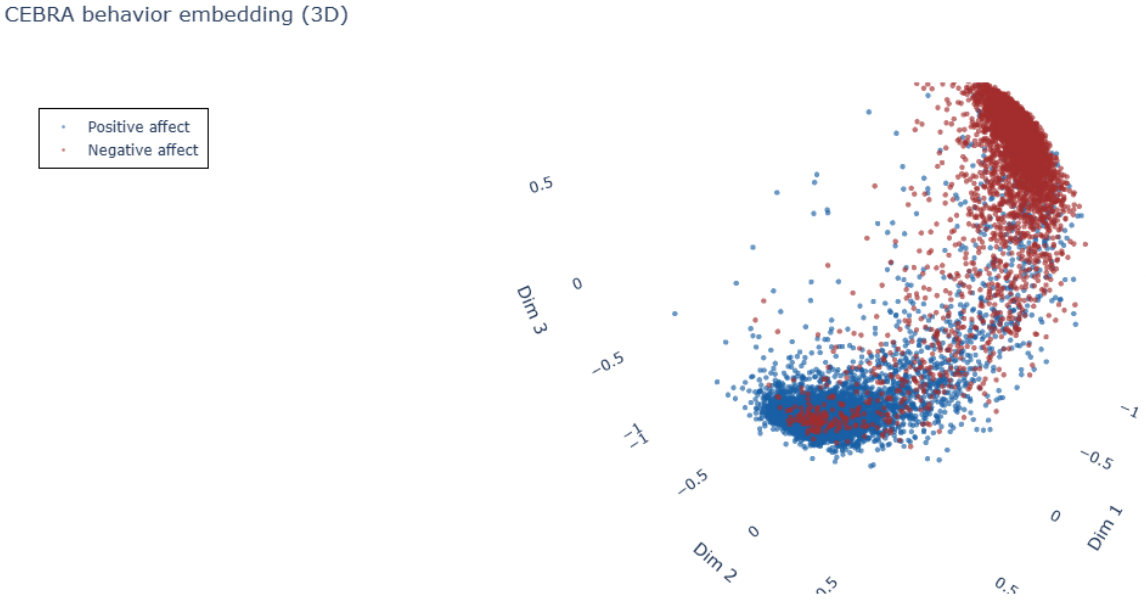



Figure 2: 3D CEBRA embedding of the dyadic signal (label-based). The embedding shows partial separation with overlapping regions, indicating a structured but non-binary representation.

### 2.4 Control Experiments

Two control experiments were performed:

**Label shuffle:** Labels were randomly permuted while keeping the signal unchanged. Accuracy dropped to approximately 0.50 across multiple runs ($0.499 \pm 0.007$), consistent with chance.

**Temporal shuffle:** The temporal structure of the signal was disrupted while preserving its distribution. Accuracy again dropped close to chance (0.515).

These controls indicate that decoding depends on structured signal rather than trivial statistics. However, because the label-shuffle and temporal-shuffle results are similar, temporal alignment cannot be isolated as the dominant factor from k-NN accuracy alone.

### 2.5 Frequency and Spatial Analysis

Band-limited analysis showed that:

- delta (1–4 Hz) achieved high decoding accuracy (0.973),
- gamma (30–45 Hz) also showed elevated performance (0.931),
- theta, alpha, and beta bands were lower in comparison.

While this suggests that low- and high-frequency components carry information, interpretation remains ambiguous. Delta may reflect slow drift or speech-envelope alignment, while gamma may include muscle-related activity. These bands cannot be interpreted as purely neural without additional controls.

Channel-level ablation indicated the strongest contribution from frontal (+0.450) and temporal regions (+0.417 left, +0.378 right), with a moderate contribution from parietal (+0.261) and a minimal contribution from occipital regions (+0.065). These regions overlap with both speech-related activity and artifact sources, so their interpretation is uncertain.

### 2.6 Summary and Limitations of the Preliminary Analysis

This analysis demonstrates that:

- joint embeddings can capture structure in time-aligned dyadic EEG;
- the model relies on structured signal rather than trivial statistics.

At the same time, several limitations remain:

- results are based on a single dyad and may not generalize;
- the ICA rejection rate is unusually high and may remove neural signal;
- delta and gamma dominance may reflect artifacts rather than neural processes;
- hybrid model performance is unstable across runs;
- within-segment normalization introduces label leakage and must be avoided;
- embedding structure may depend partly on segmentation rather than on continuous dynamics.

These issues motivate further work on stability, validation across multiple dyads, and more careful interpretation of frequency-specific results.

## 3 Methodology for Extension to Multiple Dyads

### 3.1 Dyadic Representation and Data Construction

The central design choice in this study is to treat the interacting pair as a single system rather than as two independent signals. Given synchronized EEG recordings from two participants, the signals are aligned using shared trigger markers and converted into a joint representation.

Let $\mathbf{X}_A, \mathbf{X}_B \in \mathbb{R}^{C \times T}$ denote the preprocessed signals of the two participants. After the reference channel is removed and alignment is ensured, the data are concatenated along the channel axis and transposed to form a matrix:

$$\mathbf{X} \in \mathbb{R}^{T \times 128}$$

This representation preserves temporal correspondence while allowing the model to access both participants simultaneously. No explicit assumptions are made about how the two brains interact; the embedding is expected to capture whatever joint structure is present in the data.

### 3.2 Preprocessing Pipeline and Generalization Across Dyads

The preprocessing steps used in the preliminary analysis are extended into a consistent pipeline that can operate across multiple dyads. This includes:

- alignment using trigger markers,
- removal of reference channels,
- bad-channel detection and interpolation,
- ICA-based artifact removal,
- channel-wise normalization.

Channel-wise normalization is applied after concatenation. Within-segment normalization was found in the preliminary analysis to introduce label leakage (forcing segment means to zero), and is therefore avoided.

A key issue is that preprocessing decisions that work for one dyad may not transfer directly to others. For example, the high ICA rejection rate observed in the preliminary analysis may not generalize and needs to be monitored carefully. The pipeline therefore includes basic quality checks (e.g., retained spectral structure, channel variance) to avoid over-aggressive cleaning.

The goal is not to enforce identical preprocessing across all dyads, but to define a consistent procedure with minimal manual intervention.

### 3.3 Representation Learning with CEBRA

The joint signal $\mathbf{X}$ is used to train a CEBRA model that maps each time point to a low-dimensional embedding:

$$f : \mathbb{R}^{128} \to \mathbb{R}^k$$

Three training configurations are considered:

- time-based contrastive learning (using temporal proximity),

- label-based learning (using available behavioral segmentation),
- a hybrid setup combining both.

The preliminary results indicate that the hybrid configuration can achieve higher decoding accuracy, but also that it varies across runs. Training consistency therefore needs careful evaluation before conclusions are drawn from the embeddings.

The embedding dimension, batch size, and temperature are tuned empirically rather than fixed in advance. In addition, consistency across runs is evaluated by comparing embeddings trained with different random seeds:

$$H(P, Q) \quad \text{where } P, Q \text{ are embeddings from independent runs}$$

This assesses whether the learned structures are stable or depend on initialization.

### 3.4 Evaluation Strategy and Generalization

The learned embeddings are evaluated using simple and interpretable metrics:

- k-NN decoding accuracy,
- goodness-of-fit ($R^2$) via regression from embeddings to labels,
- consistency across multiple training runs.

To assess generalization across dyads, a Leave-One-Dyad-Out (LODO) cross-validation scheme is used. The model is trained on $N - 1$ dyads and evaluated on the held-out dyad. This tests whether the embedding captures patterns that are not specific to a single pair.

In addition, learning curves are constructed by training on increasing numbers of dyads ($n = 5, 10, \dots$) to estimate how performance scales with dataset size. This shows whether the model saturates quickly or continues to improve with more data.

### 3.5 Statistical Comparison of Embeddings

Visual inspection of embeddings is not sufficient for comparison across runs or datasets. To address this, the study uses the cross-entropy test [2] as a statistical method for comparing embedding distributions.

Given two embeddings $P$ and $Q$, the test evaluates how well one distribution explains the other. In practice, it can be used to:

- compare embeddings across different training runs,
- evaluate consistency across dyads,
- test whether embeddings trained on different subsets of data have similar structure.

This component requires further implementation and validation, particularly to adapt the method to high-dimensional neural embeddings.

### 3.6 Interpretability and Frequency Analysis

The preliminary analysis showed that certain frequency bands (delta and gamma) contributed more strongly to decoding performance. However, interpretation remains ambiguous, as both bands can contain non-neural artifacts such as slow drift or muscle activity.

To better understand what the model uses, two approaches are explored:

**Frequency band analysis:** The signal is filtered into standard frequency bands and embeddings are recomputed for each band. This identifies which frequency ranges contribute to separability, but does not by itself confirm neural origin.

**Spatial contribution analysis:** Channel-wise or region-wise ablation estimates which parts of the scalp contribute most to decoding. These results are interpreted cautiously, as regions involved in speech and movement can produce similar patterns.

More advanced interpretability methods (e.g., gradient-based attribution) may also be explored, but the primary focus remains on simple and reproducible analyses.

### 3.7 Robustness and Control Experiments

To ensure that the learned embeddings reflect meaningful structure, several control experiments are included:

**Label shuffle:** Randomizing labels while keeping the signal fixed tests whether decoding relies on the true signal–label correspondence.

**Temporal shuffle:** Disrupting temporal alignment tests whether the model depends on time-locked structure. In the preliminary analysis, both controls produced similar chance-level performance, so temporal alignment cannot be isolated as the dominant factor from decoding accuracy alone.

**Phase randomization (exploratory):** Randomizing phase while preserving the power spectrum checks whether temporal structure, rather than spectral content alone, drives decoding. This requires careful implementation and is included as an additional validation step.

These controls are necessary to rule out trivial explanations such as static correlations or preprocessing artifacts.

### 3.8 Alignment Across Dyads (Exploratory)

When extending to multiple dyads, embeddings from different runs may not be directly comparable because of rotation or scaling differences.

As an exploratory step, alignment methods such as Procrustes analysis are evaluated to bring embeddings into a common coordinate system. The usefulness of such alignment depends on whether consistent structure exists across dyads, which is not guaranteed and must be tested empirically.

This part of the study is optional and depends on whether stable embeddings can be obtained across multiple runs.

# 4 Implementation and System Design

## 4.1 Overall Design Philosophy

The implementation is structured as a reproducible pipeline rather than a collection of loosely connected scripts. The main constraint is that preprocessing and embedding decisions should remain consistent across dyads, while still allowing for differences in data quality.

The workflow is organized around three stages: preprocessing and synchronization, representation learning, and evaluation. This separation is deliberate. In dyadic EEG, small preprocessing differences can propagate into large changes in the learned embedding, so each stage needs to be independently inspectable.

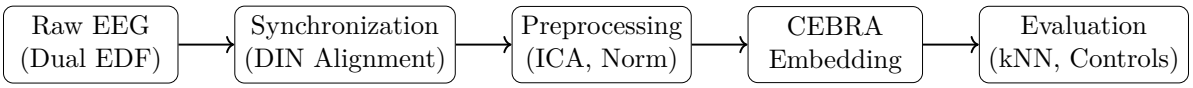



Figure 3: Pipeline overview from raw dyadic EEG to embedding evaluation. Each stage is modular and independently verifiable.

## 4.2 Software Stack and Dependencies

The system is implemented in Python using a small set of stable libraries. MNE-Python is used for EEG handling and preprocessing, including ICA. CEBRA, with a PyTorch backend, is used for representation learning. NumPy and SciPy handle numerical operations, while Pandas is used for metadata and bookkeeping.

The stack is intentionally minimal: adding more dependencies increases fragility without necessarily improving the core pipeline.

## 4.3 Repository Structure and Workflow

The repository separates reusable logic from execution code:

Listing 1: Repository structure

```
dyadic-eeg/
|-- data/            # raw and processed data (DVC tracked)
|-- src/             # preprocessing, alignment, utilities
|-- scripts/         # pipeline execution (per stage)
|-- notebooks/       # analysis and visualization
|-- environment.yml
`-- README.md
```

The `src/` directory contains all core logic. Scripts call these functions in a fixed order, and notebooks are used only for inspection and plotting. This avoids the common issue of results depending on notebook state.

## 4.4 Data Handling and Scaling Across Dyads

Moving from one dyad to many introduces two separate problems: memory usage and variability across recordings.

Memory is handled using lazy loading through `mne.io.read_raw_edf(preload=False)`, so only the required segments are loaded. This avoids loading all recordings into memory at once.

Variability is more difficult. Different dyads may have different bad channels, noise levels, or small timing inconsistencies. Instead of assuming uniform preprocessing, the pipeline includes basic checks after each stage.

| **Stage** | **Check** | **Purpose** |
|---|---|---|
| Post-normalization | Channel variance distribution | Detect flat or unstable channels |
| Post-ICA | Power spectrum shape | Verify the signal is not over-cleaned |
| Segmentation | Segment duration match | Ensure alignment across participants |

These checks are simple, but without them it is easy to propagate errors silently across multiple dyads.

### 4.5 Execution Pipeline

The processing steps are fixed and run sequentially:

| **Stage** | **Input** | **Output** |
|---|---|---|
| Synchronization | Raw EDF files | Time-aligned segments |
| Preprocessing | Aligned signals | Cleaned ($T \times 128$) tensor |
| Embedding | Tensor + labels | Low-dimensional representation |
| Evaluation | Embedding | Metrics and control results |

Each stage is independent, which makes it easier to debug issues such as misalignment or unstable embeddings without rerunning everything.

### 4.6 Handling Model Variability

The preliminary analysis showed that the hybrid CEBRA configuration varies across random seeds. This has direct implications when scaling to multiple dyads.

Instead of reporting a single number, experiments are repeated across multiple seeds and summarized by their mean and variance. If a configuration shows high variability, it is treated cautiously and compared against simpler baselines.

This avoids attributing differences between dyads to structure in the data when they may arise from training instability.

### 4.7 Data Versioning and Experiment Tracking

As the number of dyads increases, intermediate data become difficult to manage. DVC (Data Version Control) is used to track processed tensors without storing them directly in Git.

This allows preprocessing outputs to be versioned and linked to specific experiments. If a preprocessing change affects results, the exact conditions under which a result was obtained can be traced and reproduced.

Experiment settings (hyperparameters, seeds, dataset splits) are also logged explicitly rather than relying on implicit defaults.

### 4.8 Anticipated Issues

Several issues are expected when extending the pipeline.

Table 3: Technical risks and mitigation strategies for multi-dyad scaling

| Category | Constraint | Potential Impact | Mitigation Approach |
|---|---|---|---|
| Data Sync | Trigger misalignment | Loss of temporal correspondence between participants | Alignment using shared DIN markers; verify segment consistency |
| Data Sync | Small timing offsets | Reduced effectiveness of temporal contrastive learning | Cross-check alignment via signal overlap or cross-correlation |
| Preprocessing | Variable channel quality | Inconsistent signal across dyads | Channel-wise inspection and interpolation where necessary |
| Preprocessing | Aggressive ICA rejection | Possible removal of neural signal along with artifacts | Monitor spectral structure post-ICA; adjust thresholds if needed |
| Memory | Large EDF files | Slow loading and high RAM usage | Lazy loading using `preload=False` and segment-wise processing |
| Modeling | Training variability across seeds | Unstable embedding quality | Run multiple seeds and report mean/variance instead of single values |
| Modeling | Label imbalance | Biased decoding performance | Balanced sampling or stratified splits where applicable |
| Data Quality | Differences in SNR across participants | One participant dominating joint representation | Channel-wise normalization and inspection of variance |
| Interpretation | Artifact vs. neural signal ambiguity | Misleading conclusions from frequency analysis | Use controls (shuffle tests) and cautious interpretation |
| Embedding | Non-comparable latent spaces across runs | Difficulty comparing dyads directly | Explore alignment methods (e.g., Procrustes) as an optional step |
| Environment | Dependency/version mismatch | Non-reproducible results | Fixed environments via `environment.yml` or a container setup |

### 4.9 Reproducibility

The pipeline is designed so that the same inputs produce the same outputs under fixed conditions. This is achieved through explicit preprocessing steps, fixed seeds where applicable, and separation between data, code, and results.

The intent is that another user can apply the same pipeline to a new dyad and obtain comparable outputs without modifying the implementation.

## 5 Goals of the Extended Study

The goal of the extended study is a reproducible and extensible pipeline for learning and evaluating representations from dyadic EEG data. The emphasis is on ensuring that each stage (synchronization, preprocessing, embedding, and evaluation) is well defined and behaves consistently when extended from a single dyad to multiple pairs.

### 5.1 Core Components

The primary outcome is a working pipeline that takes dual-subject EEG recordings and produces validated low-dimensional embeddings. It comprises:

- a synchronization and segmentation module based on shared trigger markers,
- a preprocessing pipeline using MNE (including ICA and channel handling),
- a CEBRA-based embedding stage supporting time-based and label-based training,
- an evaluation module implementing decoding metrics and control experiments (label and temporal shuffling).

In addition to producing embeddings, the pipeline explicitly tracks how results depend on preprocessing choices and training variability. This is treated as part of the output rather than as an implementation detail.

### 5.2 Validation and Analysis

Beyond basic decoding performance, the study aims to understand how stable and comparable the learned embeddings are.

Learning curves show how performance changes as more dyads are included, and a Leave-One-Dyad-Out setup tests whether the model captures patterns that generalize across pairs.

The cross-entropy test is implemented as a statistical tool for comparing embeddings across runs or subsets of data. It is used to measure consistency, rather than assuming that embeddings from different runs are directly comparable.

A further focus is on identifying what the model uses. Frequency-restricted analysis and channel-level ablation examine whether decoding is driven by specific frequency bands or spatial regions. These results are interpreted cautiously, especially where artifact contamination is possible.

### 5.3 Reproducibility and Usability

The implementation is organized so that the full pipeline can be run end-to-end on a new dyad with minimal modification. Intermediate outputs (aligned signals, processed tensors, embeddings) are versioned and linked to specific experiment configurations.

The documentation includes a small set of notebooks demonstrating:

- preprocessing and alignment,
- embedding training under different configurations,
- evaluation and control experiments.

### 5.4 Exploratory Direction

A secondary objective is to examine whether embeddings learned from different dyads exhibit any consistent structure.

If such structure exists, simple alignment methods (e.g., Procrustes) can be used to bring embeddings into a common space. This is not assumed to work in advance, but it provides a way to test whether joint representations can be compared across interactions.

This part is exploratory and depends on the stability of the embeddings observed in earlier stages.

## 6 Planned Stages of Work

The extended study proceeds in the following stages, building a stable and verifiable pipeline first and then scaling and analyzing it across multiple dyads.

1. **Synchronization and preprocessing.** Implement and verify trigger-based alignment so that corresponding segments from both participants are time-locked, and check whether the fixed-offset correction used in the preliminary analysis generalizes across recordings. The preprocessing pipeline covers channel handling, interpolation, ICA-based artifact removal, and normalization, while avoiding within-segment normalization. The output is a reliable procedure for constructing ($T \times 128$) tensors from raw EDF inputs.
2. **Baseline embedding and evaluation.** Train and compare CEBRA configurations (time-based, label-based, hybrid), with the aim of understanding how each setup behaves under repeated runs rather than maximizing accuracy. Evaluation includes k-NN decoding, goodness-of-fit, control experiments, and explicit measurement of variability across random seeds.

3. **Scaling to multiple dyads.** Extend the pipeline to multiple dyads, test generalization with Leave-One-Dyad-Out validation, and build learning curves by incrementally increasing the number of training dyads. Practical issues such as data loading, caching, and execution time are addressed at this stage.

4. **Embedding comparison with the cross-entropy test.** Review the cross-entropy test in its original form to understand its assumptions, then adapt it to compare embedding distributions across training runs and across subsets of dyads.

5. **Interpretability and analysis.** Retrain embeddings on band-filtered signals and use channel- or region-level ablation to assess spatial contributions, interpreting low- and high-frequency effects cautiously. If embeddings are stable across dyads, test whether a shared structure exists using alignment methods such as Procrustes analysis.

6. **Consolidation and documentation.** Structure the codebase so that the full pipeline runs without manual intervention, generate all figures and tables from saved outputs, and document the pipeline with instructions and example notebooks.

## References


[1] Catherine J. Crompton et al. "Neurotype matching, not individual deficit, predicts communication success in autism". In: *Nature Human Behaviour* (2025).

[2] Carlos P. Roca et al. "A cross entropy test allows quantitative statistical comparison of t-SNE and UMAP representations". In: *Cell Reports Methods* 3.1 (2023), p. 100390. DOI: 10.1016/j.crmeth.2022.100390.

[3] Steffen Schneider et al. "CEBRA: Contrastive Embedding for Behavioral and Neural Analysis". In: *Nature* 623 (2023), pp. 103–110. DOI: 10.1038/s41586-023-06031-6.