MULTI-INTEGRATION OF LABELS ACROSS CATEGORIES FOR COMPONENT IDENTIFICATION (MILCCI)

Anonymous authors
Paper under double-blind review

ABSTRACT

Many fields collect large-scale temporal data through repeated measurements ('trials'), where each trial is labeled with a set of metadata experimental variables. These metadata often include labels spanning several categories. For example, a trial in a neuroscience study is linked to a value from category (a): task difficulty, and category (b): animal choice. A critical challenge in time-series analysis is thus to understand how these labels are encoded within the multi-trial observations, and disentangle the distinct effect of each label entry across categories. Here, we present MILCCI, a novel data-driven method that i) identifies the interpretable components underlying the data, ii) captures cross-trial variability, and iii) integrates label information to understand each category's representation within the data. MILCCI extends a sparse per-trial decomposition that leverages label similarities within each category to enable subtle, label-driven cross-trial adjustments in component compositions and to distinguish the contribution of each category. MILCCI also learns each component's corresponding temporal trace, which evolves over time within each trial and varies flexibly across trials. We demonstrate MILCCI's performance through both synthetic and real-world examples, including voting patterns, online page view trends, and neuronal recordings.

1 INTRODUCTION

A key step in understanding high-dimensional, temporally evolving systems (e.g., the brain) is analyzing time-series data from multiple repeated observations (hereafter *'trials'*). Each trial is typically labeled with a set of experimental metadata variables. Typically, such metadata span multiple *categories*; for example, each trial in neuronal recordings can be labeled with an attribute from category (a) task difficulty, and from category (b) animal's choice; each trial in weather measurements can be a time series of temperature over a day, labeled with (a) city, (b) humidity level, and (c) precipitation. We therefore refer to a trial's *label* as the tuple of its category values, e.g., '(easy task, correct choice)' or '(New York, 90% humidity, 1" snow)'. Notably, different trials can have similar or distinct labels. When one *label entry* (e.g., task difficulty) changes between trials (e.g., easy vs. challenging task), other entries (e.g., correct vs. incorrect choice) may remain the same or change as well.

Given a collection of trials in a real-world dataset, interpreting how such multi-trial, multi-label (*'multi-way'*) data varies across the space of labels is complicated by the data's high-dimensionality and variability across trials. High-dimensional data, however, can typically be represented by a smaller number of *'components'* that capture the key patterns driving observed variability. Analyzing the data in this low-dimensional space can reveal underlying trends and dynamics that capture the data's behavior across categories and the interactions amongst labels.

Existing methods for analyzing multi-way data often factorize a single, large tensor into components; however, they typically overlook trial labels and require constraints on the data structure (e.g., a tensor formed of equal-length trials). Applying these factorizations to each trial separately (via matrix factorization or other

dimensionality reduction) can account for the irregularity in data structure, but then overlooks cross-trial relationships.

Hence, there is a need for new flexible yet interpretable methods to (1) expose the underlying structure within high-dimensional multi-way data, (2) reveal how it captures label information, and (3) disentangle the effect of each category. This, in turn, demands leveraging the trial-to-trial relationships captured by the labels and understanding how these relationships govern the overall observation dynamics.

In this paper, we present **MILCCI**, a novel method to uncover the underlying structure of multi-way time-series data and disentangle how multi-category labels are embedded within it, both structurally and temporally. Our contributions include:

- We introduce MILCCI, a flexible model that exposes interpretable sparse components underlying multi-way data and reveals how they capture diverse label categories.
- MILCCI learns how the components capture label-driven variability across trials and how their activations evolve within individual trials—thus capturing the full spectrum of trial-by-trial variability.
- We validate MILCCI on synthetic data, showing it better recovers true components than baselines.
- We further demonstrate MILCCI's ability to uncover interpretable, meaningful patterns in real data, including the discovery of voting trends across US states that match actual events, patterns of online activity reflecting language and device, and neural ensembles supporting decision-making in multi-regional recordings over a thousand trials.

2 Related Work

The naive approach for analyzing multi-way data is to apply dimensionality reduction individually per trial or jointly across all trials (by stacking multiple trials into a single matrix). This can be done, e.g., via linear matrix decomposition such as PCA, ICA (Hyvarinen et al., 2001), NMF (Lee & Seung, 1999)), sparse factorization for improved interpretability (e.g., SPCA Zou et al. (2006)), or via non-linear embeddings (e.g., t-SNE (Maaten & Hinton, 2008)). However, per-trial analysis overlooks cross-trial relationships, while analyzing all trials with a single mapping ignores trial-to-trial variability in internal structure.

Demixed PCA (dPCA) (Kobak et al., 2016) isolates task-related neural variance into low-dimensional components, however it does not address missing data, different trial durations, and varying trial sampling rates, which hinders alignment across heterogeneous trials. Mudrik et al. (2024) recently introduced a unified cross-trial model that identifies building blocks encoding label information in multi-array data; however, their method handles only a single dimension of label change and thus cannot disentangle effects of multiple categories that change jointly or separately across trials. TDR and its extensions (Mante et al., 2013; Aoi & Pillow, 2018) capture multi-category labels via per-trial scalar reweighting of fixed matrices, but assume cross-trial variance arises only from linear reweighting of fixed temporal signals and cannot capture variability across trials sharing the same label.

Tensor Factorization (TF), e.g., PARAFAC (Harshman, 1970) and HOSVD (Lathauwer et al., 2000), goes beyond individual trials by treating trials as an extra data dimension in a multi-dimensional array. However, existing TF methods, including those incorporating Gaussian processes (Tillinghast et al., 2020; Xu et al., 2011; Zhe et al., 2016) or dynamic information (Wang & Zhe, 2022), are not designed to distinguish label-driven variability from other sources of variability and often produce components that are difficult to interpret. Chen et al. (2015) enable flexibility in cross-trial representations based on meta-data information, but their assumption of component orthogonality prevents the model from capturing correlated or partially overlapping patterns, which limits the representation's expressive power.

Unlike the methods above, MILCCI addresses (1) identifying the core structure of multi-trial, multi-label data with irregularities, (2) capturing trial-to-trial variability, including within repeated measures of the same label, (3) disentangling how each category is encoded within the temporal observations.

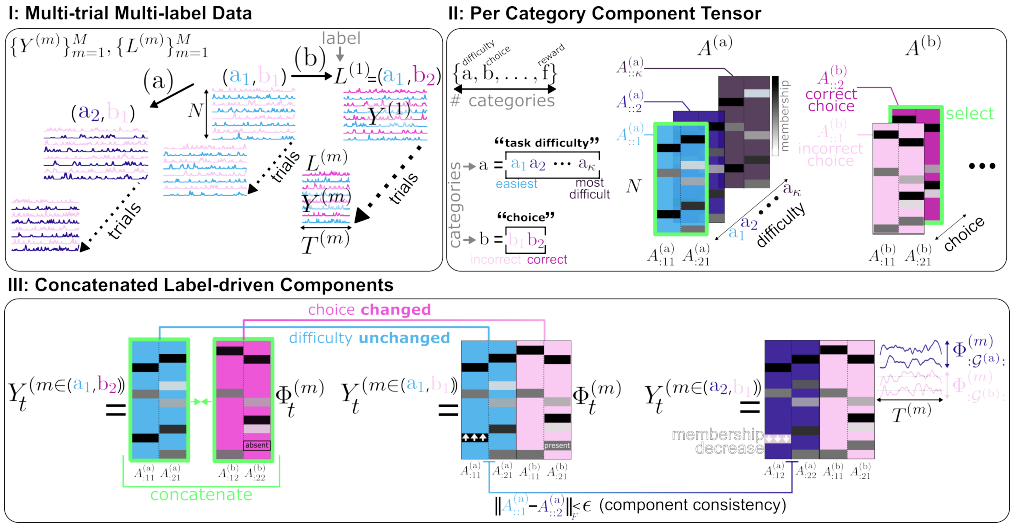

Figure 1: **Illustration.** **I:** Time-series (e.g., brain recordings) across M trials of varying duration ($\{T^{(m)}\}_{m=1}^{M}$). Each trial m is associated with a label $L^{(m)}$, which is a set of experimental variables spanning different categories (e.g., $L^{(m)} = (\text{easy task}, \text{correct choice})$). **II:** Each category (k)'s components are represented by a tensor $\mathbf{A}^{(\text{k})}$, whose i-th layer ($\mathbf{A}^{(\text{k})}_{::i}$) refers to the i-th option of that category (e.g., if the 2-nd option of category (b): correct choice, then $\mathbf{A}^{(\text{b})}_{::2}$ are correct-choice components). **III:** Each trial m is modeled via a sparse factorization, with its sparse components defined by selecting a layer from each category's tensor, based on that trial's label (green borders, **II**), and then concatenating all selected layers horizontally (green borders, **III**). This forms the loading matrix of that trial. Importantly: 1) trials with identical labels use identical loadings, 2) components can *subtly* adjust their composition under shifts in the respective category values to maintain consistency (e.g., same component under task difficulty 1 vs. 2: $\|\boldsymbol{A}^{(\text{a})}_{::1} - \boldsymbol{A}^{(\text{a})}_{::2}\|_F < \epsilon$), and 3) component temporal traces ($\{\boldsymbol{\Phi}^{(m)}\}_{m=1}^{M}$) can vary flexibly across trials.

3 Our Method: MILCCI

Problem formulation: Let $\boldsymbol{Y} = \{\boldsymbol{Y}^{(m)}\}_{m=1}^{M}$ be a set of M time-series (trials), where each $\boldsymbol{Y}^{(m)} \in \mathbb{R}^{N \times T^{(m)}}$ represents measurements from N channels across $T^{(m)}$ time points (Fig. 1, I). We assume that the N channel identities are fixed across trials, while trial durations, $T^{(m)}$, can vary. Each trial $\boldsymbol{Y}^{(m)}$ is observed under a set of trial-related metadata variables. We thus define the label of trial m as the tuple $L^{(m)}$ containing the set of each category's value in that trial (e.g., $L^{(m)} = (\text{task difficulty: easy}, \text{choice: correct})$). Notably, different trials may exhibit identical labels across all entries, partially overlapping labels, or entirely distinct labels.

To model how the labels $\{L^{(m)}\}_{m=1}^{M}$ shape the time series $\{\boldsymbol{Y}^{(m)}\}_{m=1}^{M}$, we model each $\boldsymbol{Y}^{(m)}$ as being linearly generated by a small set of P trial- and time-invariant components $\boldsymbol{A} \in \mathbb{R}^{N \times P}$. Specifically, we assume $\boldsymbol{Y}^{(m)} = \boldsymbol{A}(\boldsymbol{\Phi}^{(m)})^T + \epsilon$, where $\boldsymbol{\Phi}^{(m)} \in \mathbb{R}^{T^{(m)} \times P}$ are the components' temporal traces that define their evolution within trial m, and ϵ is *i.i.d.* Gaussian noise. We further assume that, rather than being fixed across trials, these components can undergo small adjustments in their composition between trials of different labels. For example, a brain network may recruit a few neurons during a hard task but not during an easy one. Here, our goal is to leverage the observed data $\{\boldsymbol{Y}^{(m)}\}_{m=1}^{M}$ to recover the underlying components $\boldsymbol{A}$, identify how they adjust to changes across diverse categories, and track how they evolve over time within and across trials $\{\boldsymbol{\Phi}^{(m)}\}_{m=1}^{M}$. This problem, however, is inherently challenging, as $\{\boldsymbol{Y}^{(m)}\}$ mixes all labels' effects across categories, making it hard to disentangle each category's contribution to the data.

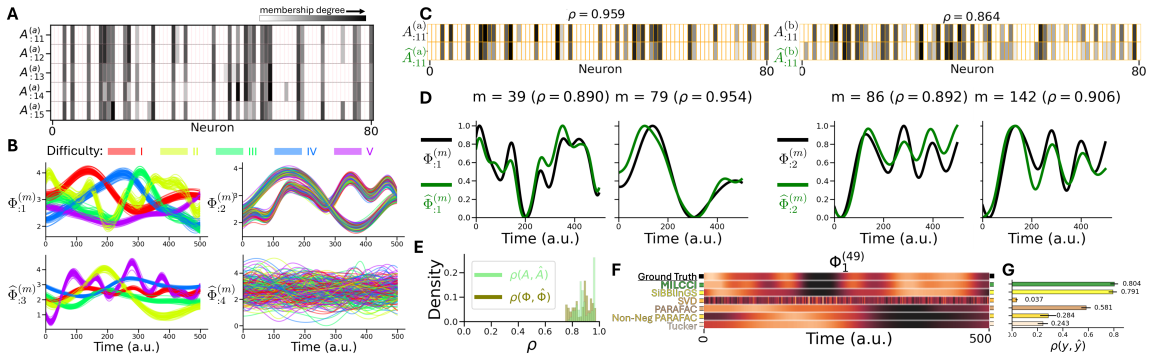

Figure 2: **MILCCI Recovers True Representations in Synthetic Data. A&B**: Generated synthetic data (full data in Fig. 5). Ground-truth components (examples in panel **A**) vary slightly across labels but remain fixed across same-label trials (rows). Ground-truth traces vary across trials (**B**, colored by difficulty). **C&D**: Identified vs. ground-truth components (top) and time-traces (bottom) for random trials. **E**: Histogram of correlations between identified components and traces vs. their true counterparts. **F&G**: Comparison of MILCCI to other methods (limited to the same 4-component dimension) based on traces (random trial, **F**) and overall reconstruction (**G**).

Our model: MILCCI learns the underlying structures from data by identifying sparse, interpretable components that adjust to changes in labels across categories. MILCCI assumes components of each category $\mathrm{k} \in \{\mathrm{a}, \mathrm{b}, \dots\}$ are captured by a category-specific sparse tensor $\boldsymbol{A}^{(\mathrm{k})} \in \mathbb{R}^{N \times p^{(\mathrm{k})} \times \#\mathrm{k}}$ (Fig. 1, II), where $p^{(\mathrm{k})}$ is the number of components and $\#\mathrm{k}$ is the number of unique options (e.g., distinct choices) in category (k).

Each tensor entry $\boldsymbol{A}^{(\mathrm{k})}_{nji}$ captures the degree of membership of the observed channel $n = 1, \dots, N$ in the j-th component under the i-th option of category (k), with 0 indicating no membership (e.g., how much neuron n participates in a specific neuronal ensemble under a correct choice). For terminology, we will denote the i-th *'layer'* of $\boldsymbol{A}^{(\mathrm{k})}$ as $\boldsymbol{A}^{(\mathrm{k})}_{::i}$, i.e., all the components corresponding to the i-th option of category (k) (e.g., correct choice components, Fig 1, II dark pink in $\boldsymbol{A}^{(\mathrm{b})}$). Notably, different tensors can have different numbers of layers, depending on how many unique values their corresponding category contains (Fig. 1, II: κ options for task difficulty vs. 2 options for choice).

For each trial $m = 1, \dots, M$ with the tuple label $\ell = L^{(m)}$, we use the tensors to build a 'dummy' matrix that captures the components to be present in that trial. From each category's tensor $\boldsymbol{A}^{(\mathrm{k})}$, we *select* a single layer i based on the category's value in ℓ (e.g., 'correct choice'). Next, we horizontally *concatenate* the matrices of all selected layers to form the 'dummy' matrix $\widetilde{\boldsymbol{A}}^{(\ell)} \in \mathbb{R}^{N \times P}$ (where $P = \sum_{(\mathrm{k})} p^{(\mathrm{k})}$), which represents trial m's components (Fig. 1, III). Notably, this creates a label-dependent (rather than time- or trial-dependent) component matrix that reuses layers from the global tensor set. While the components are defined by the label only, each j-th component is also associated with a temporally evolving trace ($\boldsymbol{\Phi}^{(m)}_{:j} \in \mathbb{R}^{T^{(m)} \times 1}$) that marks its activation within each trial m and can vary flexibly between and within trials.

This label-driven variation of component compositions over distinct-label trials allows flexibility, yet should ensure the consistency of each component across trials (e.g., if a neuronal ensemble recruits additional neurons only under a difficult task, it should still be identifiable as the 'same' ensemble under an easy task). Hence, any pair of layers $i \neq i'$ of each (k) category's tensor ($\boldsymbol{A}^{(\mathrm{k})}_{::i}, \boldsymbol{A}^{(\mathrm{k})}_{::i'}$) are constrained to be similar, but need not be strictly identical, via $\|\boldsymbol{A}^{(\mathrm{k})}_{::i} - \boldsymbol{A}^{(\mathrm{k})}_{::i'}\|_F < \delta(\mathrm{k}_i, \mathrm{k}_{i'})$ (Fig. 1, III). Here, δ is a small value reflecting the similarity between the i-th and i'-th options of category (k) (e.g., how 'close' task difficulty 1 is to task difficulty 4). This way, the degree of compositional adjustment scales with the magnitude of label change (App. A.2).

We then model the data $\boldsymbol{Y}^{(m)}$ of each trial m as lying in the low-dimensional space defined by the sparse 'dummy' matrix $\widetilde{\boldsymbol{A}}^{(\ell)}$, via $\boldsymbol{Y}^{(m)} = \widetilde{\boldsymbol{A}}^{(\ell)}(\boldsymbol{\Phi}^{(m)})^T + \epsilon$, where ϵ is *i.i.d.* Gaussian noise.

Model training: MILCCI uncovers the underlying per-category component tensors ($\{\boldsymbol{A}^{(\mathrm{a})}, \boldsymbol{A}^{(\mathrm{b})}, \ldots\}$) and their traces from the time series ($\{\boldsymbol{Y}^{(\mathrm{m})}\}_{m=1}^{M}$). To support both categorical and ordinal labels, MILCCI optionally pre-computes a label similarity graph $\boldsymbol{\lambda}^{(\mathrm{k})}$ for each (k) (App. A.2), which is used in component inference to allow the compositional adjustments to reflect label similarity. Then, we initialize the components and traces (App. A.1) and take an iterative procedure to infer $\{\boldsymbol{A}^{(\mathrm{k})}\}_{k\in\text{Categories}}$ and $\{\boldsymbol{\Phi}^{(m)}\}_{m=1}^{M}$.

Inferring $\{\boldsymbol{A}^{(\mathrm{a})}, \boldsymbol{A}^{(\mathrm{b})}, \ldots\}$:
For each unique option k_i of each category (k) (e.g., each task difficulty level), we infer the i-th layer of the tensor $\boldsymbol{A}^{(\mathrm{k})}$ ($\boldsymbol{A}^{(\mathrm{k})}_{::i}$). For this, we use all trials observed under this k_i value, regardless of the values of the other categories. Namely, we use the subset of observations $\{\boldsymbol{Y}^{(m)}\}_{m\in\widetilde{M}}$ where $\widetilde{M} = \{m \in \{1,\ldots,M\} \mid L^{(m)}_{\mathrm{k}} = \mathrm{k}_i\}$ (e.g., all trials with difficulty level = 1, regardless of choice).

Using this observations subset $\{\boldsymbol{Y}^{(m)}\}_{m\in\widetilde{M}}$ we first need to separate $\boldsymbol{A}^{(\mathrm{k})}_{::i}$'s contribution from that of the other components (coming from layers of other tensors) in the concatenated matrix $\widetilde{\boldsymbol{A}}^{(L^{(m)})}$. Hence, for each $m \in \widetilde{M}$, we first calculate the residual between $\boldsymbol{Y}^{(\mathrm{m})}$ and the contributions of all layers $\boldsymbol{A}^{(\mathrm{k'})}_{::i}$ $\forall(\mathrm{k'}) \neq (\mathrm{k})$, by: $\widetilde{\boldsymbol{Y}}^{(m)} = \boldsymbol{Y}^{(m)} - \sum_{\mathrm{k'}\neq\mathrm{k}} \boldsymbol{A}^{(\mathrm{k'})}_{::L^{(m)}_{\mathrm{k'}}}(\boldsymbol{\Phi}^{(m)}_{:,\mathcal{G}^{(\mathrm{k'})}})^T$, where $\boldsymbol{A}^{(\mathrm{k'})}_{::L^{(m)}_{\mathrm{k'}}}$ are the components from some layer of $\boldsymbol{A}^{(\mathrm{k'})}$ (category (k')'s tensor), and $\boldsymbol{\Phi}^{(m)}_{:,\mathcal{G}^{(\mathrm{k'})}} \in \mathbb{R}^{T^{(m)}\times p^{(\mathrm{k'})}}$ are the traces of these (k') category components.

We can then solve the following optimization for $\boldsymbol{A}^{(\mathrm{k})}_{::i}$ using LASSO (Tibshirani, 1996):

$$\widehat{\boldsymbol{A}}^{(\mathrm{k})}_{::i} = \arg\min_{\boldsymbol{A}^{(\mathrm{k})}_{::i}\geq 0} \|\widetilde{\boldsymbol{Y}}^{(m)} - \boldsymbol{A}^{(\mathrm{k})}_{::i}\boldsymbol{\Phi}^{(m)}_{:\mathcal{G}^{(\mathrm{k})}}\|_F^2 + \gamma_1 \sum_{i'\neq i} \lambda^{(\mathrm{k})}_{i',i}\|\boldsymbol{A}^{(\mathrm{k})}_{::i'} - \boldsymbol{A}^{(\mathrm{k})}_{::i}\|_F^2 + \gamma_2\|\mathrm{vec}(\boldsymbol{A}^{(\mathrm{k})}_{::i})\|_1, \tag{1}$$

where $\boldsymbol{A}^{(\mathrm{k})}_{::i} \geq 0$ applies an optional non-negativity constraint on the components, 'vec$(\cdot)$' flattens a matrix into a vector, $\lambda^{(\mathrm{k})}_{i',i}$ denotes the similarity regularization between the i-th and i'-th option of category (k) (App. A.2), and γ_1, γ_2 are hyperparameters promoting cross-label smoothness and sparsity, respectively. The first term thus enforces data fidelity; the second promotes consistency between corresponding components proportional to their label similarity as captured by $\boldsymbol{\lambda}^{(\mathrm{k})}$; and the third encourages component sparsity. Each component is then normalized to a fixed sum to avoid scaling ambiguity with $\boldsymbol{\Phi}^{(m)}$.

Updating $\{\boldsymbol{\Phi}^{(\mathrm{m})}\}_{m=1}^{M}$:
Since traces vary over trials, updates are applied per trial m. For each trial m with multi-category label $\ell := L^{(m)}$, we update $\boldsymbol{\Phi}^{(m)}$ by:

$$\widehat{\boldsymbol{\Phi}}^{(m)} = \arg\min_{\boldsymbol{\Phi}^{(m)}} \underbrace{\|\boldsymbol{Y}^{(\mathrm{m})} - \widetilde{\boldsymbol{A}}^{(\ell)}(\boldsymbol{\Phi}^{(m)})^T\|_F^2}_{\text{data fidelity}} + \gamma_3 \underbrace{\sum_{t=1}^{T^{(m)}} \|\boldsymbol{\Phi}^{(m)}_t - \boldsymbol{\Phi}^{(m)}_{t-1}\|_2^2}_{\text{temporal smoothness}} + \gamma_4 \underbrace{\|\mathrm{vec}((\mathbf{C}\odot(\mathbf{1}\text{–}\mathbf{I}_p))\odot \boldsymbol{D})\|_1}_{\text{de-correlation}} \tag{2}$$

where $\mathbf{C} := (\boldsymbol{\Phi}^{(m)})^T\boldsymbol{\Phi}^{(m)}$, with $\odot$ denoting element-wise multiplication, $\boldsymbol{D}_{f,i} := \|\boldsymbol{\Phi}^{(m)}_{:i}\|_2^{-1}\|\boldsymbol{\Phi}^{(m)}_{:f}\|_2^{-1}$, and γ_3, γ_4 hyper-parameters. This optimization thereby promotes data fidelity and smoothness (first and second terms), and regularizes temporal trace correlations within trials (third term). See Alg. 1 and Fig. 1.

4 Experiments

MILCCI Recovers True Components from Synthetic Data: We generated synthetic data arising from $P = 4$ sparse components with time-varying traces ($T = 500$ time points; $M = 250$ trials). We defined two

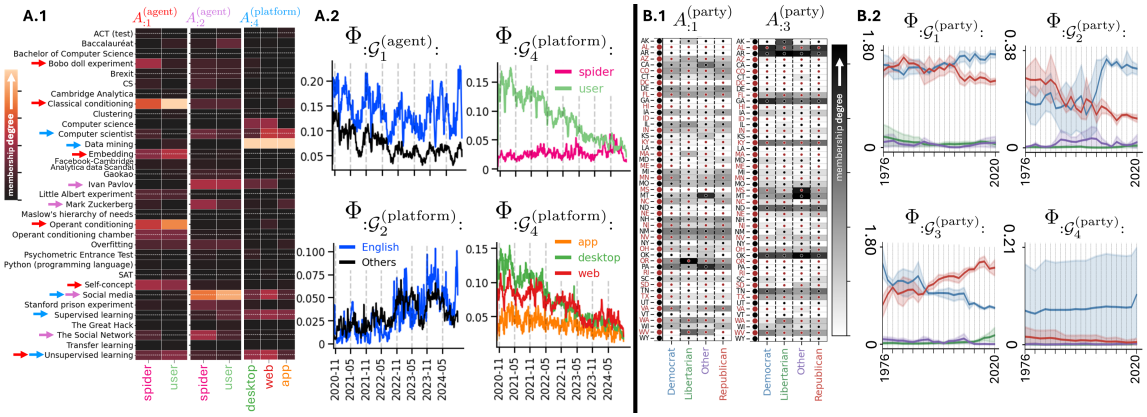

Figure 3: **Wikipedia Results (A):** Identified Wiki page-view exemplary components (**A**.1) and traces averaged by different categories (**A**.2); **Voting Results (B):** Identified exemplary components (**B**.2) and traces (**B**.3, Mean and 80% confidence interval). See full in Fig. 6

categories: (a) 'task difficulty' (5 options), and (b) 'choice' (2 options), such that $p^{(a)} = p^{(b)} = 2$ ensembles adjust with changes in (a) and (b) (Fig. 5B). Each trace was generated as a Gaussian process with parameters varying across components and trials: some reflect task difficulty, some choice, and some vary each trial (Fig. 2, B, Fig. 5, A, App. C). We ran MILCCI on this data, comparing to (1) Tucker (Tucker, 1966), (2) PARAFAC (Harshman, 1970), (3) non-negative PARAFAC, (4) SVD, and (5) SiBBlInGS (Mudrik et al., 2024), all using the same $P = 4$ components (Fig. 2, F,G). Since TF is invariant to component permutation, we resorted the components of each method using linear sum assignment to align with the ground truth.

MILCCI recovered the true components (Fig. 2, C) and traces (Fig. 2, D), with high correlations to the ground truth for both (Fig. 2, E). Compared to other methods, the representation identified by MILCCI was most similar to the ground truth (Fig. 2, F,G, green). SiBBlInGS also performs well (Fig. 2F) but produces smoother traces. Correlation of the reconstructed components with the ground truth (Fig. 2G) is higher for MILCCI than the baselines; while SiBBlInGS achieves a similar correlation, MILCCI provides greater interpretability by distinguishing the contribution of each category.

MILCCI Reveals State-Level Voting Patterns by Party and Office: Voting data consist of 50 US states $\times$ 23 years (sampling every 2 or 4 years) $\times$ party (e.g., Democrat, Republican, Libertarian; see Tab. 2) $\times$ office (Presidency, Senate, House) were extracted from (Data & Lab, 2017a;b;c) and normalized for each state (App. D). Some states lack data for certain years or office due to differing election schedules (Fig. 6).

We applied MILCCI to this data using $P = 8$ components, with categories (a) party, and (b) office ($p^{(k)} = 4$ components each). MILCCI discovers components capturing state-specific voting patterns that vary by office, party, and time. For example, component $\boldsymbol{A}_1^{(\text{party})}$ (Fig. 3, B.1) highlights Montana (MT) and Pennsylvania (PA) having increased membership in the 'Other' category, primarily driven by the Independent Party (MT, Fig. 13, right) and the Constitution Party (PA, Fig. 13, left). This aligns with MT's historical emphasis on individualism (Kitayama et al., 2010) and PA hosting the Constitution Party headquarters. The same component identifies Oregon's increased Libertarian membership, matching the 2001 law that eased ballot access for minor parties (Oregon Elections Division, n.d.) and reflecting that the Libertarian Party of Oregon was among the earliest state branches. Notably, its trace ($\boldsymbol{\Phi}_{:\mathcal{G}_1^{(\text{party})}}$, Fig. 3, B.1, top-left) shows overlapping Democrat–Republican activations diverging $\sim$2004, with Democrat activity rising and Republican activity decreasing, reflecting long-term partisan realignment driven by national political shifts (e.g., Iraq War 2003).

In component $\boldsymbol{A}^{(\text{party})}_{:3}$ (Fig. 3, B.1, right), MILCCI groups AK, OK, AL, AZ, MS, MT together. This grouping matches the legislative similarities between these states, e.g., strict voter ID laws (National Conference of State Legislatures, 2025), demonstrating MILCCI's effectiveness in recovering hidden trends directly from observations. Temporally, its trace $\boldsymbol{\Phi}_{:\mathcal{G}^{(\text{party})}_3}$ (Fig. 3, B.2, bottom-left) shows opposing trends between Democrat and Republican activations, with Republican activation rising, and Libertarian activity emerging around 2016. This trend reflects a rise in Republican votes, a decline in Democratic votes, and a possible shift of some Democratic support toward the Libertarian party in these states. These and other patterns identified by MILCCI (App. D.2) demonstrate its ability to uncover state–party–office–dependent patterns. Notably, components from other TF methods (Fig. 10,7) are dense (PARAFAC), include negative values (SVD, PARAFAC, Tucker), or, in SiBBlInGS, fail to disentangle party from office, which hinders interpretability (Fig. 9). Moreover, due to their restrictive tensor structure, PARAFAC and Tucker do not capture compositional adjustments and cannot flexibly vary their traces to capture trial-to-trial temporal variability.

MILCCI Finds Wikipedia Page Clusters Across Devices and Languages: Next, we extracted Wikipedia Pageview counts (Meta, 2022) (Oct. 20'–24', $T = 1482$, daily) across three categories: (a) agent: user/spider (i.e., web crawler), (b) platform: desktop/web/app, and (c) language: en/ar/es/fr/he/hi/zh. Each trial records the number of Pageviews under a label (e.g., $\boldsymbol{Y}^{(m)}$: Pageviews over time in English, via desktop, by a human); data pre-processing in App. E.1. We identified components that cluster related Wiki-pages together and vary across categories (Fig. 3, A), with some pages (e.g., unsupervised learning) appearing in more than one component, emphasizing MILCCI's ability to capture multi-meaning terms.

Particularly, we identified, e.g., $\boldsymbol{A}^{(\text{agent})}_1$ grouping Learning Theory (in Psychology) pages (Fig. 3, A.I, red arrows, list in App. E.2); $\boldsymbol{A}^{(\text{agent})}_2$ grouping social-media pages (purple arrows); and component $\boldsymbol{A}^{(\text{platform})}_4$ grouping computer science basics (blue arrows). Interesting patterns emerge when exploring how components composition adjust to e.g., user↔spider and desktop↔web↔app.

For $\boldsymbol{A}^{(\text{agent})}_1$ (Learning theory in Psychology), MILCCI finds small differences across agents (spider vs. user, Fig. 3, A.I, left). For instance, the Bobo Doll Experiment's Wiki-page (a psychological experiment on social learning theory) appears under 'spider' but not 'user'. This matches it being less familiar to the average person than other psychology terms in the cluster, while spiders are linked to it through the actual Wikipedia links connecting this page to other related terms. Accordingly, other Wiki-pages, like classical and operant conditioning, which are foundations in psychology, show higher membership magnitudes in user. 'Unsupervised learning' and 'embedding' also show small membership in this component, higher in 'user' than 'spider'. Interestingly, these pages refer to CS terms (not psychology), but since they also carry meaning in psychological learning, the higher 'user' membership compared to 'spider' matches human behavior: users enter the page but leave upon realizing the term refers to a different field, whereas spiders follow predictable navigation. These findings highlight MILCCI's ability to reveal distinct human vs. spider behaviors within the same component, and also underscore the importance of allowing compositional adjustments to capture nuanced trends, unlike other methods (Fig. 19).

This component's trace ($\boldsymbol{\Phi}_{:\mathcal{G}^{(\text{agent})}_1:}$, Fig. 3, A.II top-left) captures its fluctuations and higher activation in English compared to other languages. This aligns with professional terms being more elaborated in English, often using jargon not fully defined/used in non-English languages, and with non-English native speakers possibly preferring to read professional material in English (Miquel-Ribé & Laniado, 2018).

The social-media component $\boldsymbol{A}^{(\text{agent})}_2$ shows small structural adjustments user↔spider: 'Mark Zuckerberg' is higher in spider, while 'social media' is higher in user, possibly reflecting reduced human interest in figures versus common terms like social media. Its trace ($\boldsymbol{\Phi}_{:\mathcal{G}^{(\text{agent})}_2:}$) rises until a peak in Mar. 2024 in both English and non-English (Fig. 3A.II, bottom-left), matching the general increase in social media and possibly related to, e.g., Florida's Social Media Ban in Mar. 2024, which was widely noted and is mentioned on the corresponding 'social media' Wiki-page captured by this component.

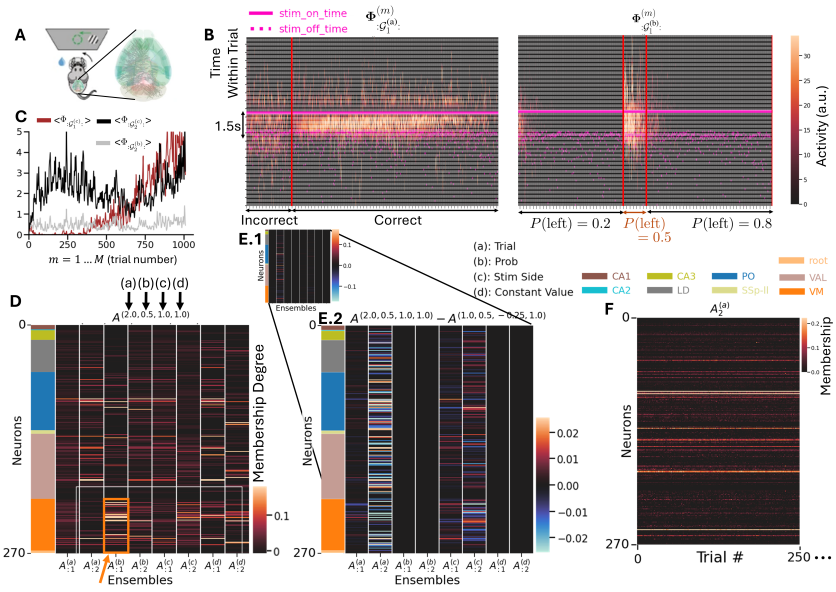

Figure 4: **MILCCI identifies meaningful neuronal ensembles in real-world brain data. A**: Experimental setting (from Angelaki et al. (2025)). **B**: Traces (all 1011 trials) sorted horizontally by trial correctness (left panel) and $p(\text{left})$ (right panel). **C**: Average within-trial values of exemplary traces reveal varying degrees of temporal drifts over trials. **D**: Ensembles identified (exemplary trial). **E**: Differences in ensemble composition across trials. **F**: Trial-adjusted ensemble compositions over first 250 trials.

$\boldsymbol{A}_4^{(\text{platform})}$ (computer science basic terms) captures membership adjustments to platform, with, e.g., computer scientist (a short Wikipedia page without math/graphs) showing higher membership under app/web compared to under desktop. This contrasts with, e.g., unsupervised learning, whose page is more complex (includes math and graphs) and shows lower membership in the app compared to the other platforms. This may match users' preference to read 'easy' terms on the app and more complex terms on desktop, and shows MILCCI's ability to reveal interpretable processing differences across platforms.

This component's temporal trace ($\boldsymbol{\Phi}_{:\mathcal{G}_4^{(\text{platform})}}$) shows higher activity in English (Fig. 14), with access dominated by desktop and mobile web (Fig. 3, A.II, bottom-right). The temporal trace is mostly driven by users rather than spiders (Fig. 3, A.II, top-right); the user trace decreases over time, aligning with these terms being basic ('old') in CS (compared to newer trends, like LLMs). This highlights MILCCI's power in discovering platform-specific engagements and behavioral fingerprints.

Some more interesting patterns include, e.g., $\boldsymbol{A}_{:2}^{(\text{platform})}$ that captures terms related to Cambridge Analytica; its trace peaks around 2020, mainly in English, and decreases since, aligning with the timeline of this case. See Figs. 14, 17, 16, 18, 19 for comparisons and full traces.

MILCCI Finds Neural Ensembles underlying Multi-Region Brain Data: Finally, we apply MILCCI to neuronal activity patterns from multi-regional, single-cell-resolution recordings of mice in a decision-making task (Laboratory et al. (2025); Angelaki et al. (2025), App. F). In this experiment mice reported the location of a visual grating with varying contrast by turning a wheel left or right (Fig. 4, A). We used data from a random session, extracted the available spike time data, and estimated firing rates by applying a Gaussian convolution. We removed inactive neurons, and split trials into trial start and end, yielding $N = 270$ neurons over 137 time points across $M = 1011$ trials (example processed trials in Fig. 21). We defined $p^{(\text{k})} = 2$ components ('neuronal ensembles') per category: (a) trial number, (b) prior side, (c) stimulus side, and (d) fixed components that are fixed across trials. This setup (1) enables distinguishing representational drift (Rule et al., 2019; Driscoll et al., 2022) potentially related to learning or attention, from task variables, and (2) demonstrates MILCCI's ability to simultaneously support both non-adjusting and adjusting components, across both categorical and ordinal categories.

We found neuronal ensembles selective for diverse task variables (Fig. 4). For example, $\mathbf{\Phi}^{(m)}_{:\mathcal{G}_1^{(a)}}$ is tuned to choice correctness, with activity surging during stimulus presentation (Fig. 4, B left) under *correct* choice only. Interestingly, MILCCI also found an ensemble sensitive to *incorrect* decisions, activated just after stimulus presentation ($\mathbf{\Phi}_{:\mathcal{G}_2^{(b)}:}$, Fig. 22). Notably, these components can provide insight as to how neuronal ensembles integrate stimulus information to support correct decision-making, and to relate these traces to the specific neurons involved (Fig. 24). In another example, $\mathbf{\Phi}^{(m)}_{:\mathcal{G}_1^{(b)}:}$ is mostly active during trials with a random-prior (i.e., $p(\text{left}) = 0.5$), throughout before, during, and after the stimulus, and is largely inactive in trials where the prior favors one side (Fig. 4, B right). This highlights MILCCI's efficiency in suggesting the involvement of priors in decision-making, and its capability of exposing hidden similarities within traces of similar conditions.

Moreover, MILCCI's ability to isolate components that track representational drift ($\mathbf{\Phi}^{(m)}_{:\mathcal{G}_1^{(c)},:}$, Fig. 4, C brown; 23) reveals how neuronal coding evolves over trials, which can be the result of learning, attention, or adaptation. MILCCI reveals how ensemble compositions across regions (Fig. 4, D; colors on the left mark brain regions of neurons) capture distinct local structures. For example, $\boldsymbol{A}^{(b)}_{:1}$ captures many neurons in the primary visual area (VM, orange arrow in Fig. 4D), suggesting it may be involved in visual stimulus processing. We can also identify interesting patterns via the ensemble compositions (Fig. 4, D), and how they change between trials (e.g., between two exemplary trials with differing stimulus sides; Fig. 4E.1, zoom-in on changes in Fig. 4E.2). These reveal patterns in ensemble composition adjustments: the first ensemble of $\boldsymbol{A}^{(a)}$ (Fig. 4E.2, left column) adjusts minimally, while the second ensemble exhibits distributed adjustments across areas. In stimulus-adjusting ensembles (5-th and 6-th columns), adjustments occur with some localization around VM (e.g., $\boldsymbol{A}^{(c)}_{:2}$, Fig. 4), suggesting an adaptive VM composition tracking stimulus side.

5 Discussion, Limitations, and Future Steps:

Here, we presented MILCCI, a data-driven method for analyzing multi-trial, multi-label time series. MILCCI (i) identifies interpretable components underlying the data, (ii) captures cross-trial relationships, and (iii) integrates label information, while accounting for trial-to-trial variability beyond label effects. By allowing components to adjust their composition with label changes, MILCCI uncovers similarities that could remain hidden under fixed-component factorizations. Unlike tensor factorization approaches, MILCCI maintains interpretability while modeling cross-trial variability, integrating labels, and disentangling label-driven from non-label-driven sources of variation. Another strength of MILCCI, in contrast to existing methods, is its ability to simultaneously handle multiple label types within a single dataset, including categorical, non-continuous ordinal, continuous, and trial-varying labels (e.g., as in the IBL experiment).

We validated MILCCI on synthetic data, where it outperformed alternative methods, and demonstrated its effectiveness in diverse real-world settings: (1) exposing voting patterns aligned with real-world events; (2) recovering interpretable components in Wikipedia view data and tracking how memberships vary across languages, platforms, and agents; and (3) revealing neuronal ensembles that adapt their composition across trials and task variables, including ensembles tuned to decision correctness and prior information.

One potential concern is the scaling ambiguity between components and traces, which MILCCI addresses by normalizing components after each iteration while allowing traces to flexibly vary across trials to capture trial-specific amplitudes. Although component dimension per category is a hyperparameter, MILCCI naturally handles this, as sparsity drives redundant components to zero, which mitigates hyperparameter sensitivity. The current approach assumes linear decomposition within each trial, which ensures interpretability, with exciting opportunities to extend it to nonlinear relationships. Another exciting direction for future work is to extend the non-directional components into directional ones that capture interactions between member channels, for example, by introducing a dynamics prior during inference (via e.g.,Chen et al. (2024); Linderman et al. (2016)).

REFERENCES

Dora Angelaki, Brandon Benson, Julius Benson, Daniel Birman, Niccolò Bonacchi, Kcénia Bougrova, Sebastian A Bruijns, Matteo Carandini, Joana A Catarino, et al. A brain-wide map of neural activity during complex behaviour. *Nature*, 645(8079):177–191, 2025.

Mikio Aoi and Jonathan W Pillow. Model-based targeted dimensionality reduction for neuronal population data. *Advances in Neural Information Processing Systems*, 31, 2018.

Po-Hsuan (Cameron) Chen, Janice Chen, Yaara Yeshurun, Uri Hasson, James Haxby, and Peter J Ramadge. A reduced-dimension fmri shared response model. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Yenho Chen, Noga Mudrik, Kyle A Johnsen, Sankaraleengam Alagapan, Adam S Charles, and Christopher Rozell. Probabilistic decomposed linear dynamical systems for robust discovery of latent neural dynamics. *Advances in Neural Information Processing Systems*, 37:104443–104470, 2024.

MIT Election Data and Science Lab. U.S. President 1976–2020, 2017a. URL https://doi.org/10.7910/DVN/42MVDX.

MIT Election Data and Science Lab. U.S. House 1976–2022, 2017b. URL https://doi.org/10.7910/DVN/IG0UN2.

MIT Election Data and Science Lab. U.S. Senate statewide 1976–2020, 2017c. URL https://doi.org/10.7910/DVN/PEJ5QU.

Laura N Driscoll, Lea Duncker, and Christopher D Harvey. Representational drift: Emerging theories for continual learning and experimental future directions. *Current opinion in neurobiology*, 76:102609, 2022.

Richard A Harshman. oundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16(1):1–84, 1970.

Aapo Hyvarinen, J Karhunen, and E Oja. Independent component analysis and blind source separation, 2001.

Shinobu Kitayama, Lucian Gideon Conway III, Paula R Pietromonaco, Hyekyung Park, and Victoria C Plaut. Ethos of independence across regions in the united states: The production-adoption model of cultural change. *American Psychologist*, 65(6):559, 2010.

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

International Brain Laboratory, Brandon Benson, Julius Benson, Daniel Birman, Niccolò Bonacchi, Matteo Carandini, Joana Catarino, Gaelle Chapuis, Peter Dayan, Eric DeWitt, Tatiana Engel, Michele Fabbri, Mayo Faulkner, Ila Fiete, Charles Findling, Laura Freitas-Silva, Berk Gerçek, Kenneth Harris, Sonja Hofer, Fei Hu, Félix Hubert, Julia Huntenburg, Anup Khanal, Christopher Langdon, Petrina Lau, Guido Meijer, Nathaniel Miska, Jean-Paul Noel, Kai Nylund, Alejandro Pan-Vazquez, Alexandre Pouget, Cyrille Rossant, Noam Roth, Rylan Schaeffer, Michael Schartner, Yanliang Shi, Karolina Socha, Nicholas Steinmetz, Karel Svoboda, Anne Urai, Miles Wells, Steven West, Mathew Whiteway, Olivier Winter, and Ilana Witten. Ibl - brain wide map [deprecated] (version draft). https://dandiarchive.org/dandiset/000409/draft, 2025. Data set, DANDI Archive.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21:1253–1278, 2000. URL https://api.semanticscholar.org/CorpusID:14344372.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Scott W Linderman, Andrew C Miller, Ryan P Adams, David M Blei, Liam Paninski, and Matthew J Johnson. Recurrent switching linear dynamical systems. *arXiv preprint arXiv:1610.08466*, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pp. 689–696, 2009.

Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, 2013.

Meta. Pageviews analysis — meta, discussion about wikimedia projects, 2022. URL https://meta.wikimedia.org/w/index.php?title=Pageviews_Analysis&oldid=23610159. [Online; accessed 31-October-2024].

Marc Miquel-Ribé and David Laniado. Wikipedia culture gap: quantifying content imbalances across 40 language editions. *Frontiers in physics*, 6:54, 2018.

Noga Mudrik, Gal Mishne, and Adam Shabti Charles. Sibblings: Similarity-driven building-block inference using graphs across states. In *Forty-first International Conference on Machine Learning*, 2024.

National Conference of State Legislatures. Voter identification laws. https://www.ncsl.org/elections-and-campaigns/voter-id, 2025. Updated June 2, 2025.

Oregon Elections Division. *Political Party Manual*. Oregon Secretary of State, 255 Capitol St NE, Suite 126, Salem, OR 97310-0722, n.d. URL https://sos.oregon.gov/elections/Documents/PoliticalPartyManual.pdf. Oregon Administrative Rule No. 165-010-0005.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL http://jmlr.org/papers/v12/pedregosa11a.html.

M. Ravasi and I. Vasconcelos. Pylops–a linear-operator python library for scalable algebra and optimization. *SoftwareX*, 11:100495, 2020. doi: https://doi.org/10.1016/j.softx.2020.100495. URL https://www.sciencedirect.com/science/article/pii/S2352711020302449.

Oliver Rübel, Andrew Tritt, Ryan Ly, Benjamin K Dichter, Satrajit Ghosh, Lawrence Niu, Pamela Baker, Ivan Soltesz, Lydia Ng, Karel Svoboda, et al. The neurodata without borders ecosystem for neurophysiological data science. *Elife*, 11:e78362, 2022.

Michael E Rule, Timothy O'Leary, and Christopher D Harvey. Causes and consequences of representational drift. *Current opinion in neurobiology*, 58:141–147, 2019.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Conor Tillinghast, Shikai Fang, Kai Zhang, and Shandian Zhe. Probabilistic neural-kernel tensor decomposition. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 531–540. IEEE, 2020.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

Ewout Van Den Berg and Michael P Friedlander. Probing the pareto frontier for basis pursuit solutions. *Siam journal on scientific computing*, 31(2):890–912, 2009.

Zheng Wang and Shandian Zhe. Nonparametric factor trajectory learning for dynamic tensor decomposition. In *International Conference on Machine Learning*, pp. 23459–23469. PMLR, 2022.

Zenglin Xu, Feng Yan, et al. Infinite tucker decomposition: Nonparametric bayesian models for multiway data analysis. *arXiv preprint arXiv:1108.6296*, 2011.

Shandian Zhe, Kai Zhang, Pengyuan Wang, Kuang-chih Lee, Zenglin Xu, Yuan Qi, and Zoubin Ghahramani. Distributed flexible nonlinear tensor factorization. *Advances in neural information processing systems*, 29, 2016.

Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

Appendix

A Running Details

A.1 Initialization:

We initialize the components and traces using dictionary learning Mairal et al. (2009) with sparsity on the components, using Sklearn.decomposition (Pedregosa et al., 2011) DictionaryLearning. Within the model, sparsity is applied through PyLops Ravasi & Vasconcelos (2020) SPGL1's solver Van Den Berg & Friedlander (2009).

A.2 Details about state similarity graph calculation:

MILCCI allows disentangling of categorical, continuous, and non-continuous ordinal categories. It further supports allowing the components to adjust across trials with label change in a way that captures the degree of similarity between the corresponding labels for each category. For example, if we assume neuronal ensembles gradually present compositional shifts over the course of learning, this requires capturing the gradual / ordinal order of trials. Another example is when a certain category represents a continuous variable (e.g., x-position of a stimulus), where again we would like to capture label relationships (i.e., distance between labels).

Hence, MILCCI augments the model with a set of label-driven graphs that are pre-calculated before the beginning of the iterative optimization process and are reused across iterations for smoother cross-component regularization that maintains label similarity. For each category (k), we build the graph $\boldsymbol{\lambda}^{(\mathrm{k})} \in \mathbb{R}^{\#\mathrm{k}\times\#\mathrm{k}}$, where #k is the number of unique options observed under category (k) (e.g., if category "choice" can be correct / wrong, then #k=2). This graph captures the degree of similarity between its possible values.

For categorical labels, we use a constant value for the graph (e.g., $\lambda^{(\mathrm{k})}_{i,i'} = 1 \ \ \forall i, i'$).

For ordinal labels: We use a Gaussian kernel $\lambda^{(\mathrm{k})}_{i,i'} = e^{\frac{\|\mathrm{k}_i - \mathrm{k}_{i'}\|_2^2}{2\sigma^2}} \ \ \forall i, i'$, where k_i and $\mathrm{k}_{i'}$ are the i-th and i'-th option of category (k) (e.g., task difficulty 1 vs. 5). Notably, MILCCI supports integration of diverse graph calculation distance metrics, so one can easily use a different distance metric (not Gaussian) if they assume similarities between labels are captured differently.

After the graph calculation, we recommend normalizing the graph by the per-row absolute sum of 1 to ensure that different labels are regularized to the same degree:

$$\lambda^{(\mathrm{k})}_{i,:} \leftarrow \frac{\lambda^{(\mathrm{k})}_{i,:}}{\|\lambda^{(\mathrm{k})}_{i,:}\|_1} \ \ \forall i, i'.$$

An i-th row of zeros in $\boldsymbol{\lambda}^{(\mathrm{k})}$ means that the i-th option of category (k) is not regularized to be consistent with the others. This can be used if there is some intention to create completely trial-varying components that vary flexibly between trials, which is another feature MILCCI offers.

B Versions

We trained the model using Python 3.10.4 (conda-forge) with matplotlib 3.8.2, scikit-learn 1.0.2, seaborn 0.11.2, numpy 1.23.5, pandas 1.5.0, PyLops 1.18.2, and SPGL1 0.0.2.

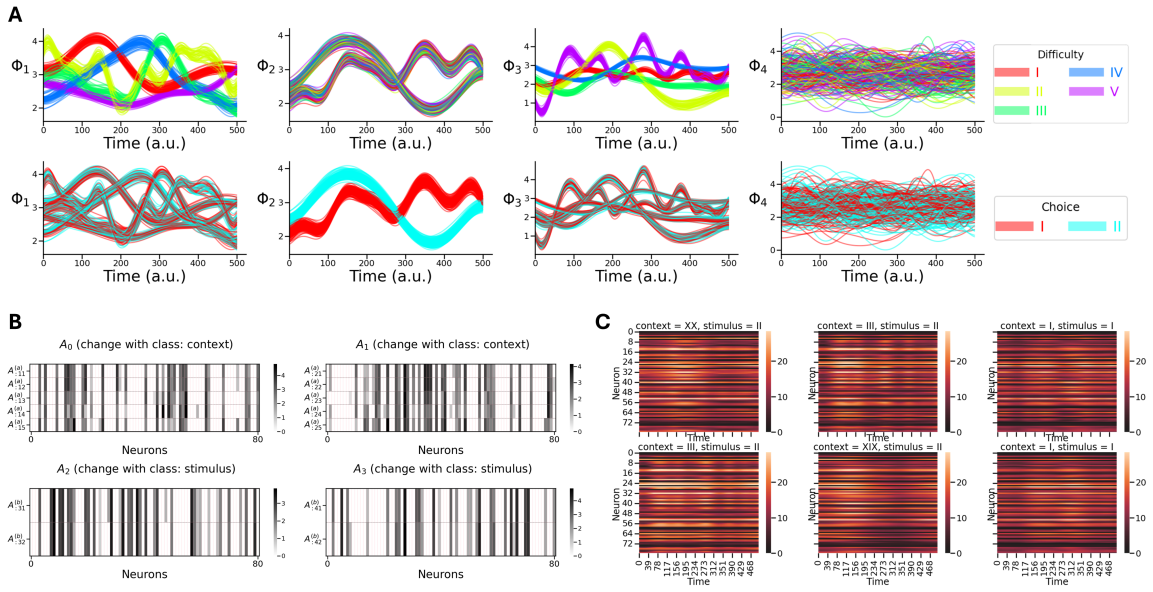

Figure 5: **Generated Synthetic Data. A:** Generated traces, colored by difficulty (top) or choice (bottom). **B:** Generated components. Each subplot shows one component and how it varies over the labels of each category (changes across rows). In other words, each subplot corresponds to $\boldsymbol{A}^{(\mathrm{k})}_{:j:}$ for some component j. **C:** Random exemplary generated synthetic trials $\{\boldsymbol{Y}^{(m)}\}$.

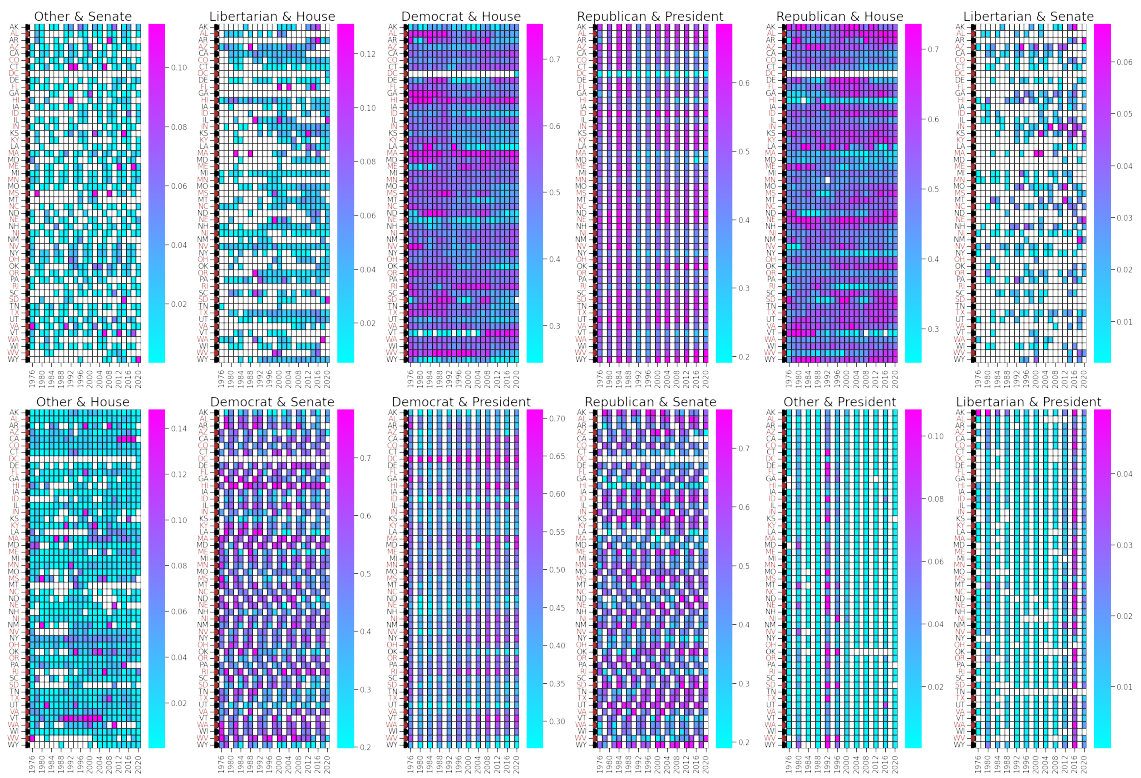

Figure 6: **Voting Data**: ($\{\boldsymbol{Y}^{(m)}\}_{m=1}^{M}$) Show Diverse Data Structures

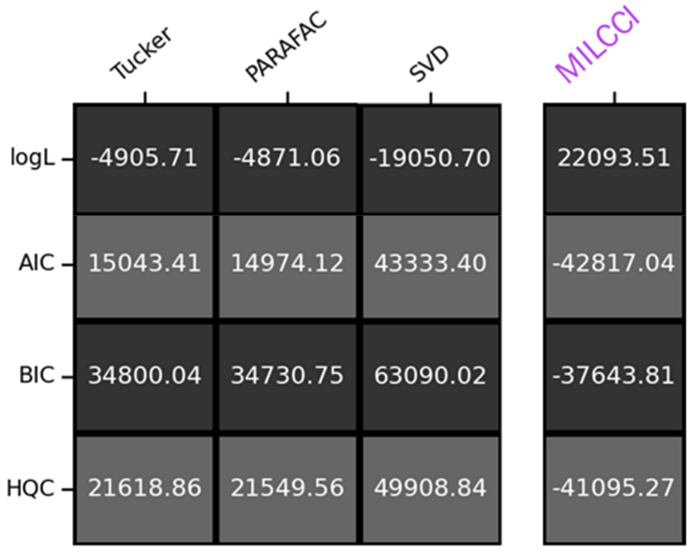

Figure 7: **Voting Experiment.** comparison to baselines in reconstruction and information criteria. Comparison to Tucker, PARAFAC, and SVD in terms of reconstruction and information criteria, including 1) log-likelihood of the observations given the identified components, and 2) information criteria that balance reconstruction and model complexity: AIC (Akaike Information Criterion), BIC (Bayesian Information Criterion), and HQC (Hannan-Quinn Criterion). Lower values indicate better performance.

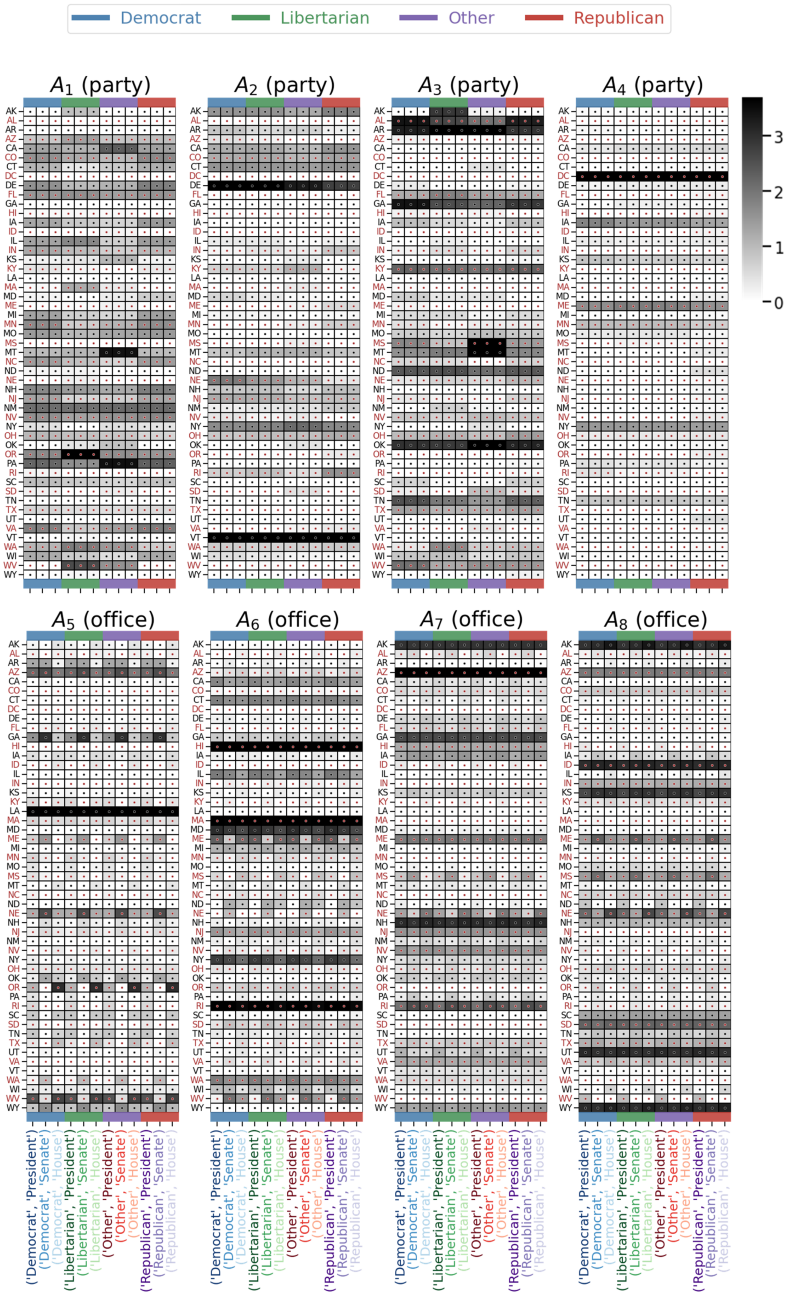

Figure 8: Identified Ensembles for Voting Experiment.

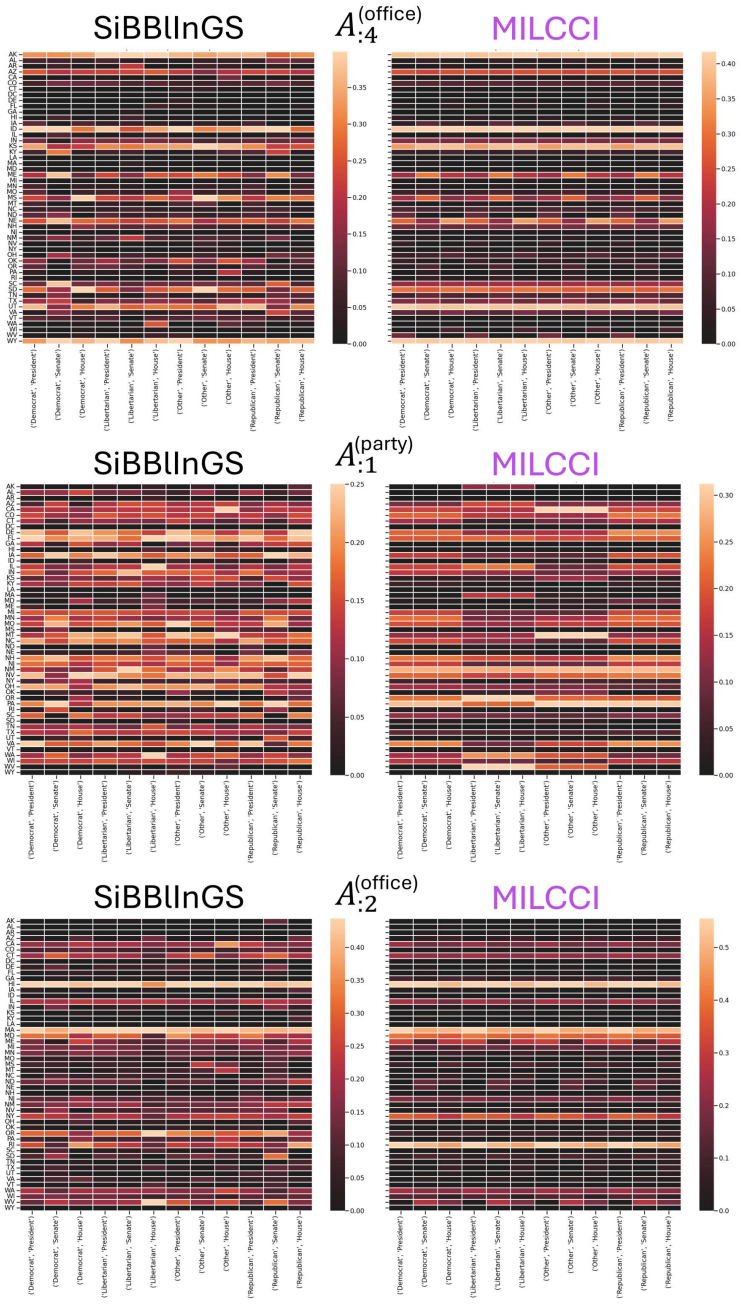

Figure 9: Comparison of components identified by MILCCI and SiBBlInGS for three exemplary labels under the same parameters and seed. MILCCI components were duplicated to match the x-tick labels of SiBBlInGS. SiBBlInGS shows uninterpretable changes across every label, even when parts are shared, whereas MILCCI disentangles them.

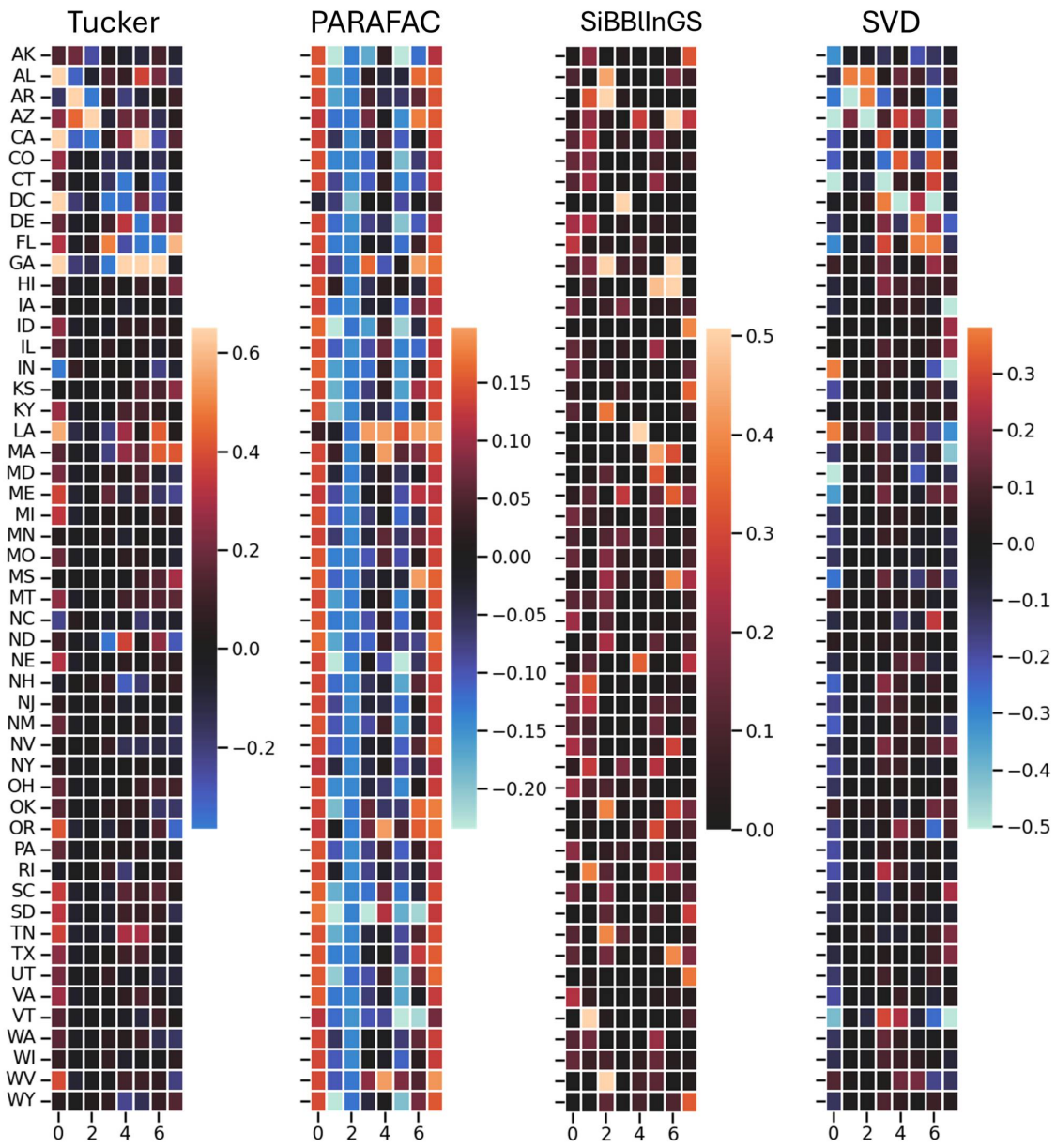

Figure 10: **Voting Experiment Baseline Comparison**. Components identified by the following baselines: 1) Tucker Decomposition (HOSVD), 2) PARAFAC, 3) SiBBlInGS (for a single random label layer: (Democrat, President)), 4) SVD (on all concatenated trials).

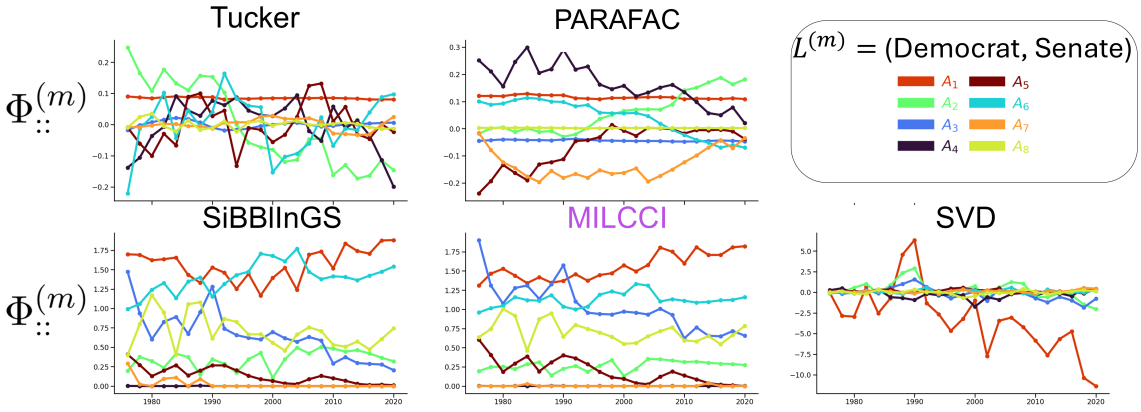

Figure 11: **Voting Experiment**. Traces Identified by MILCCI compared to the other baselines for exemplary random trial.

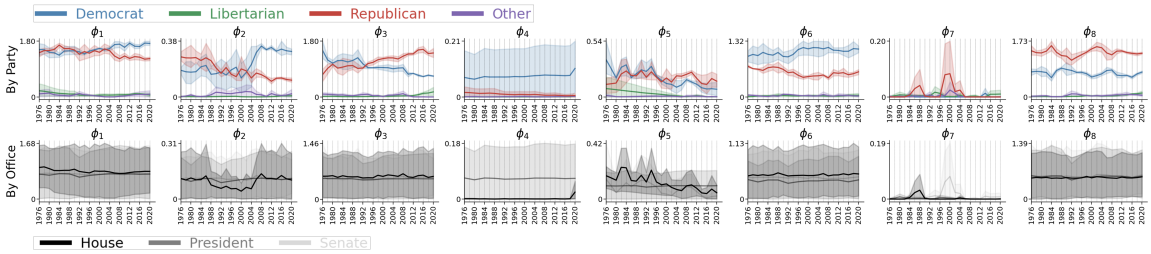

Figure 12: **Voting Traces.** Top: Colored by Party. Bottom: Colored by Office.

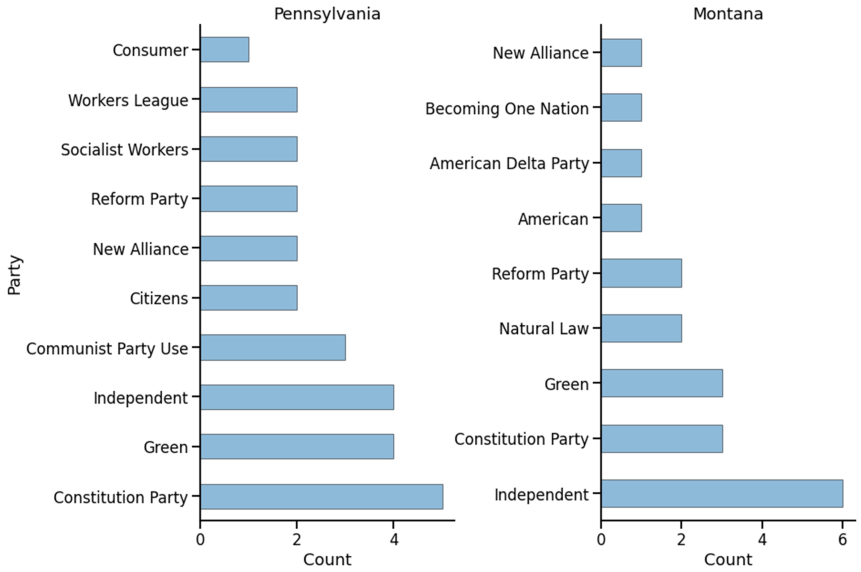

Figure 13: Top "Other" parties instance counts for Montana and Pennsylvania

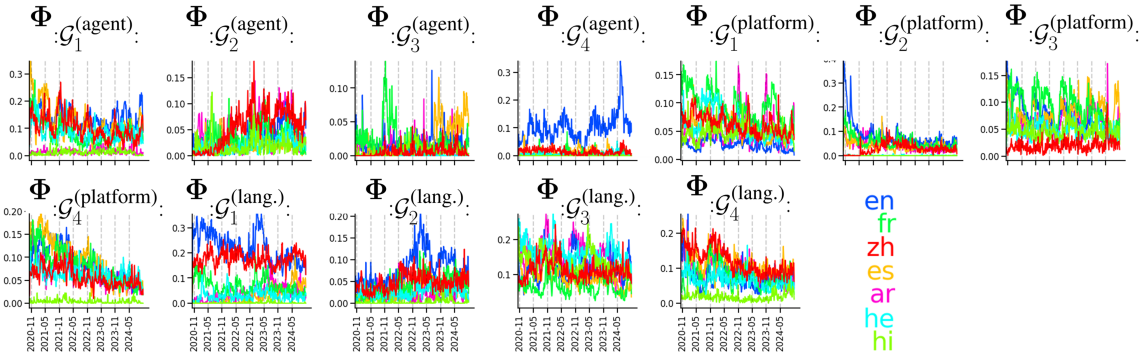

Figure 14: **Wikipedia Page View Experiment**. Traces colored by Language (Arabic, English, Espagnole, French, Hebrew, Hindi, Chinese) Across All Ensembles.

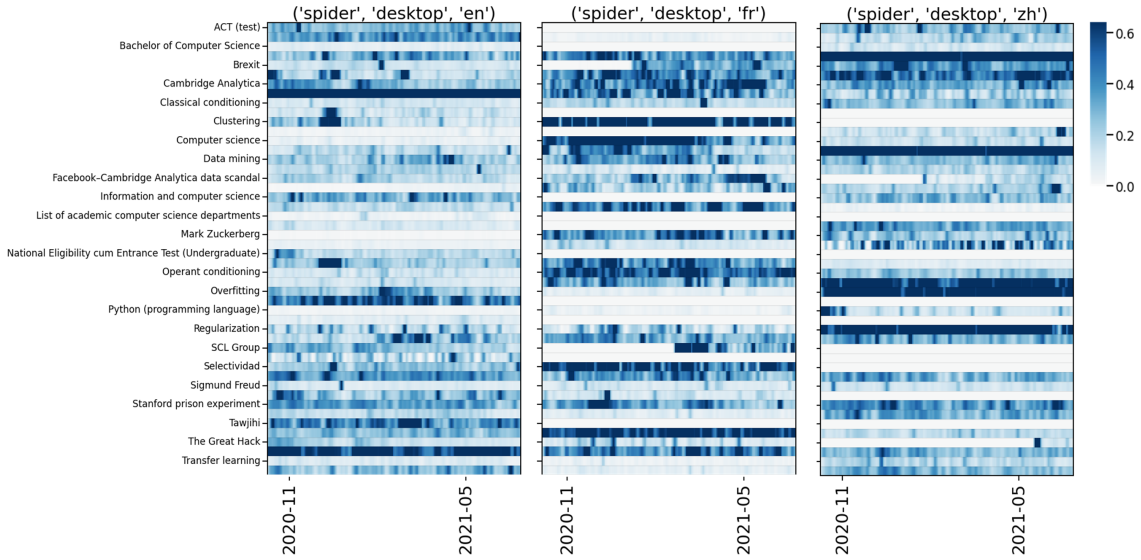

Figure 15: Wikipedia Page View Data, Exemplary Trials

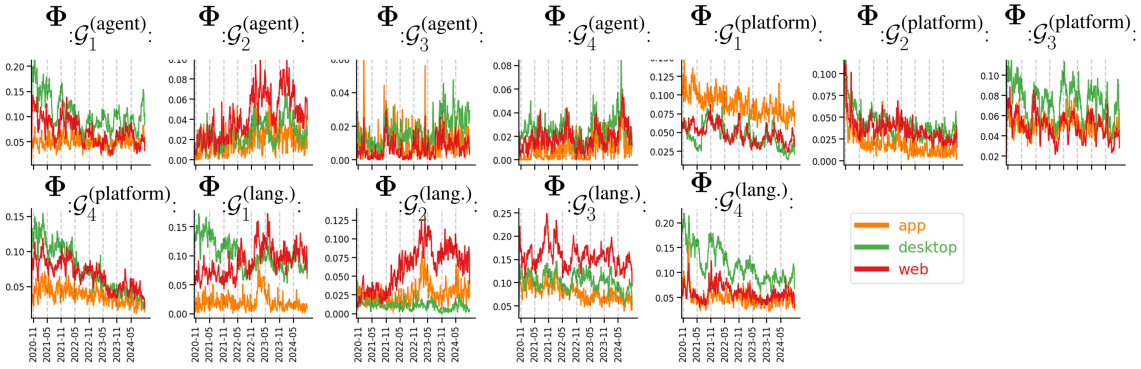

Figure 16: **Wikipedia Page View Experiment**. Traces colored by Platform (desktop, mobile web, mobile app) Across All Ensembles

Algorithm 1 MILCCI Algorithm

Input: Observed trial data $\{\boldsymbol{Y}^{(m)}\}_{m=1}^{M}$, with associate multi-category labels $\{L^{(m)}\}_{m=1}^{M}$.
Pre-calculate: Label-similarity graph $\boldsymbol{\lambda}^{(\mathrm{k})}$ for each category (k) (App. A.2).
Initialize: Sparse components $\{\boldsymbol{A}^{(\mathrm{k})}\}_{\mathrm{k}\in\text{Categories}}$ and traces $\{\boldsymbol{\Phi}^{(m)}\}_{m=1}^{M}$ ▷ App. A.1
while not converged **do**
  for each category (k) **do**
    for each label value k_i **do**
      Compute residuals for trials with label k_i
      Solve for $\boldsymbol{A}^{(\mathrm{k})}_{::i}$ with cross-label consistency and sparsity via LASSO ▷ equation 1
      Normalize each component to sum to 1 ($\boldsymbol{A}^{(\mathrm{k})}_{:ri} \leftarrow \frac{\boldsymbol{A}^{(\mathrm{k})}_{:ri}}{\|\boldsymbol{A}^{(\mathrm{k})}_{:ri}\|_1}$) ▷ to prevent scaling ambiguity with Φ
  for each trial m with label ℓ **do**
    Build the stacked component matrix $\widetilde{\boldsymbol{A}}^{(\ell)}$ by selecting a layer from each $\boldsymbol{A}^{(\mathrm{k})}$.
    Update traces $\boldsymbol{\Phi}^{(m)}$ to minimize data fidelity, smoothness and de-correlation ▷ equation 2

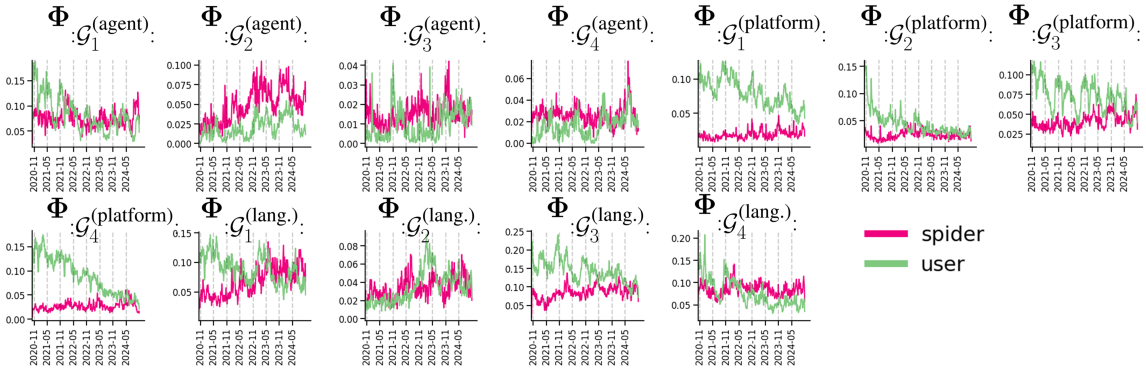

Figure 17: **Wikipedia Page View Experiment**. Traces colored by agent (Spider vs. User) Across All Ensembles

C Additional Information—Synthetic Experiment

We generated synthetic datasets with **80 channels**, **4 components** (2 categories, $\times$ 2 components adjusting per category), **500 time points per trial**, and **250 trials**. Each trial received one label per axis: category (a) (difficulty)'s labels were sampled from the set $\{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{VI}, \mathrm{V}\}$ (5 levels) and category (b) (choice) labels from $\{\mathrm{I}, \mathrm{II}\}$ (2 levels), yielding up to **10 unique label combinations**. Component-to-neuron maps were initialized for the reference trial with values in $[0.5, 1.0]$, then updated across label layers according to a trial-similarity graph calculated based on trial labels and thresholded at the **60th percentile** to enforce sparsity on the component compositions.

Temporal activity for each label pair was drawn from a Gaussian-process prior with an RBF kernel scaled by a per-sample amplitude: the amplitude was drawn per sample in $\approx [0.2, 1.533]$, and the kernel length scale was drawn per sample in $[0.05, 0.2]$ in normalized time units (0–1), which corresponds roughly to **25–100 time points** given 500 samples per trial. A white-noise term of 1×10^{-8} was included in the kernel. For each label we drew one GP sample and then generated multiple similar trial traces by adding multivariate-

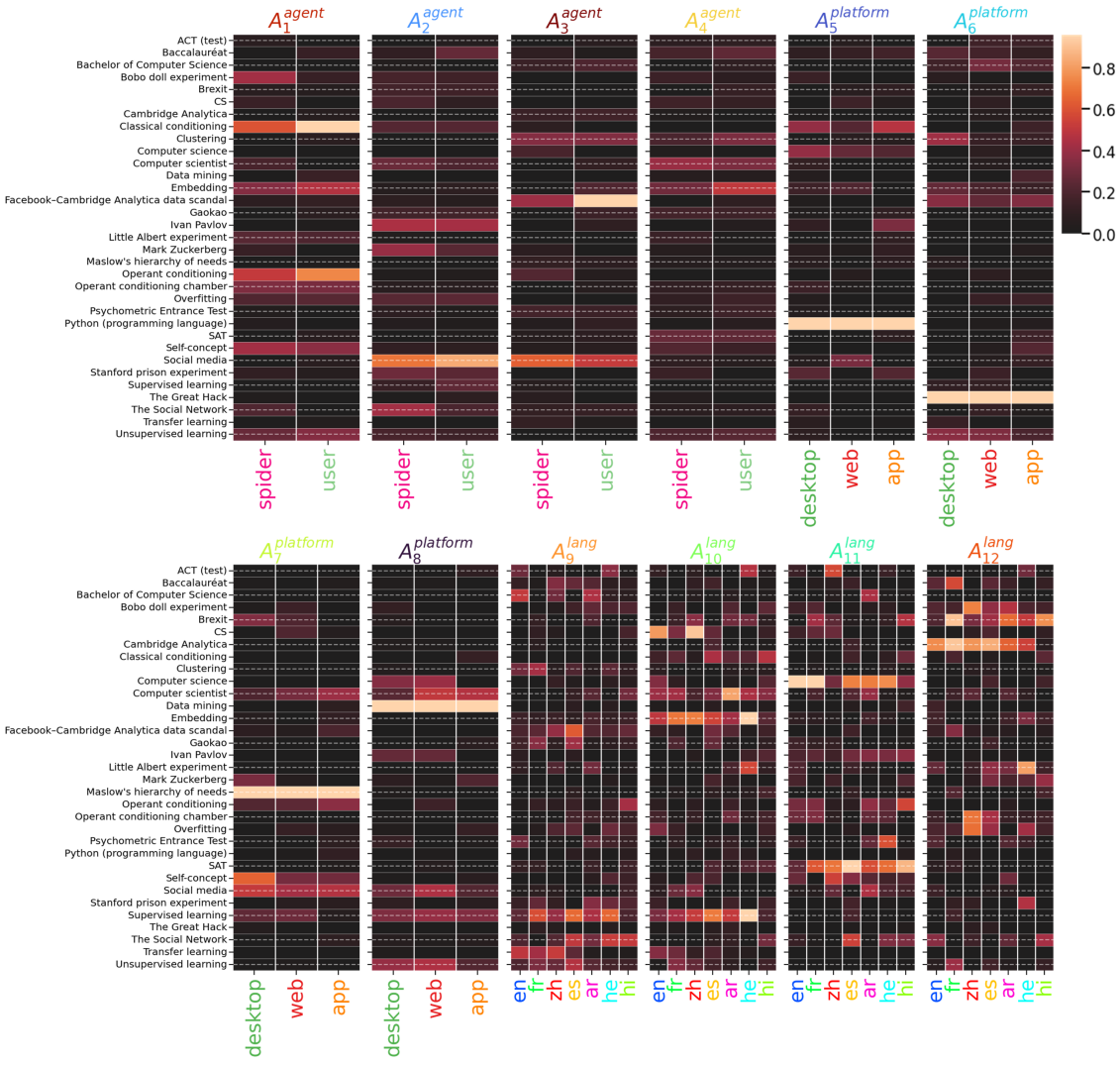

Figure 18: Components identified for Wikipedia Page-view Experiment.

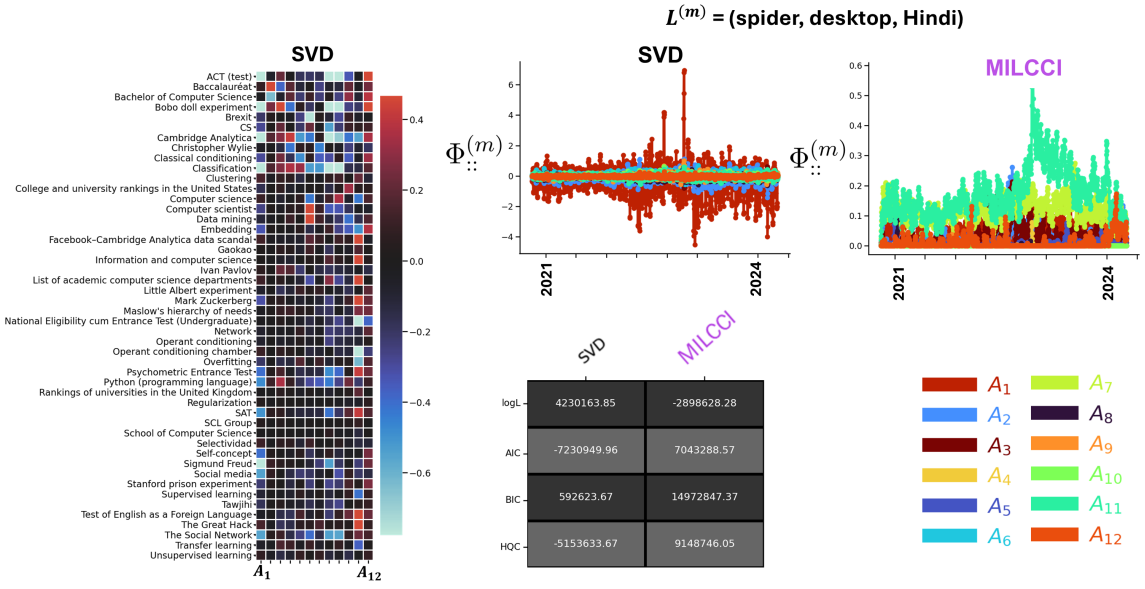

Figure 19: **Wikipedia Experiment Compared to SVD**. Components identified by SVD (compositions on the left, example trial traces on the right). Only SVD is shown here, as PARAFAC and Tucker (implemented via PyLops Ravasi & Vasconcelos (2020)) did not converge (see App. G). SiBBlInGS components Mudrik et al. (2024), which are more complex, are presented separately in Fig. 20.

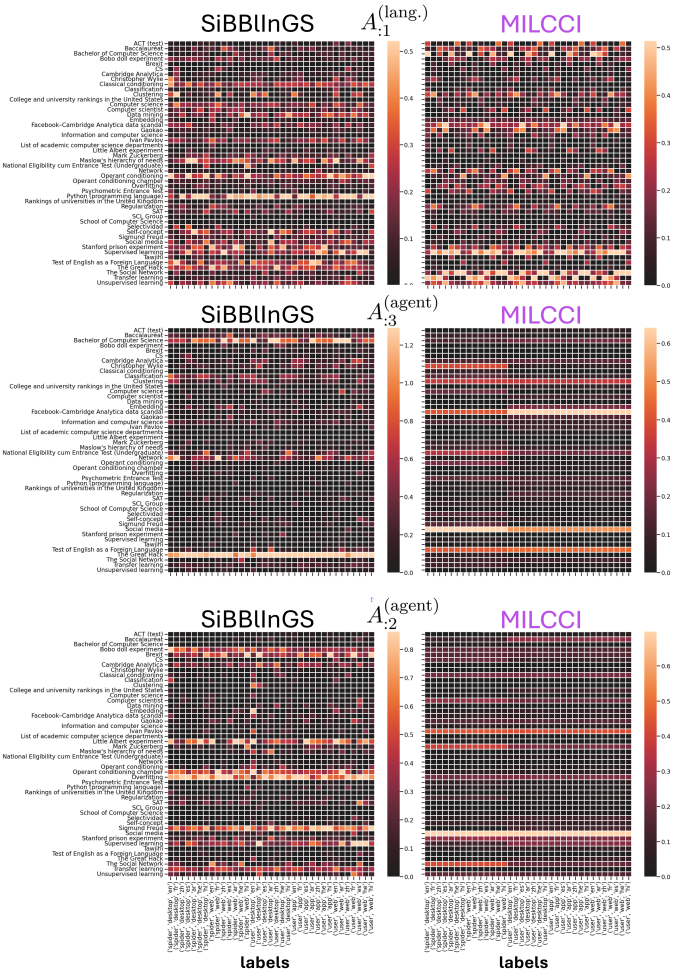

Figure 20: **Wikipedia Components, MILCCI vs. SiBBlInGS**. SiBBlInGS components display compositional changes scattered across labels, rather than the category-specific adjustments captured by MILCCI. *Note:* MILCCI components are shown here with duplicate columns to align with SiBBlInGS components for visualization.

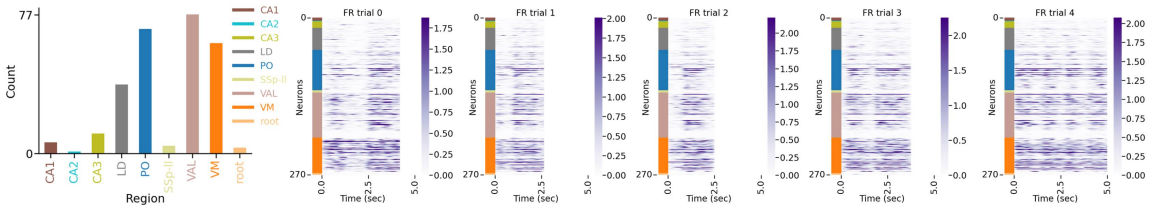

Figure 21: IBL data following our pre-processing steps.

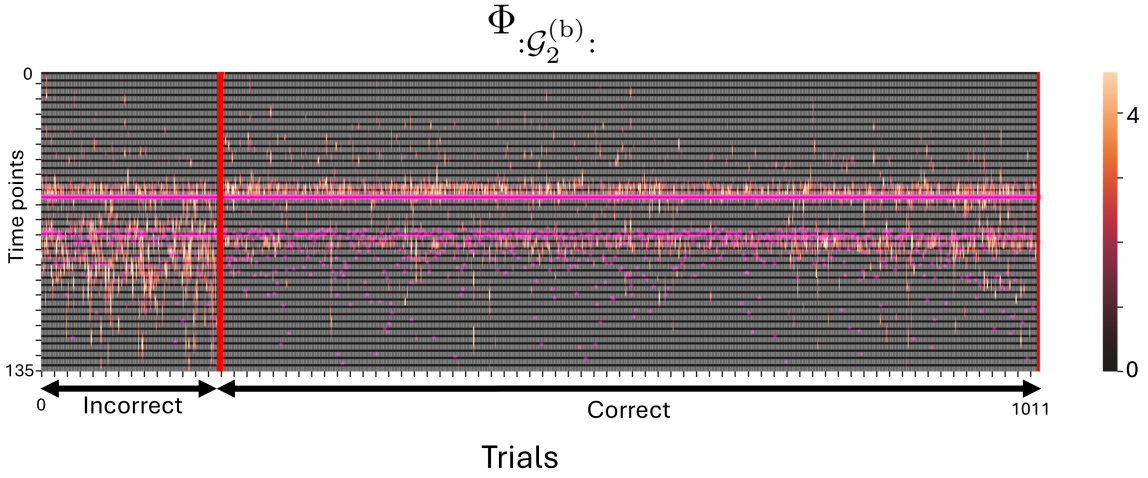

Figure 22: Traces of $\mathbf{\Phi}_{:\mathcal{G}_2^{(b)}:}$ across trials, separated by decision correctness. Solid pink: stimulus on. Dashed pink: stimulus off. The stimulus appears for a median duration of 1.45 s across trials.

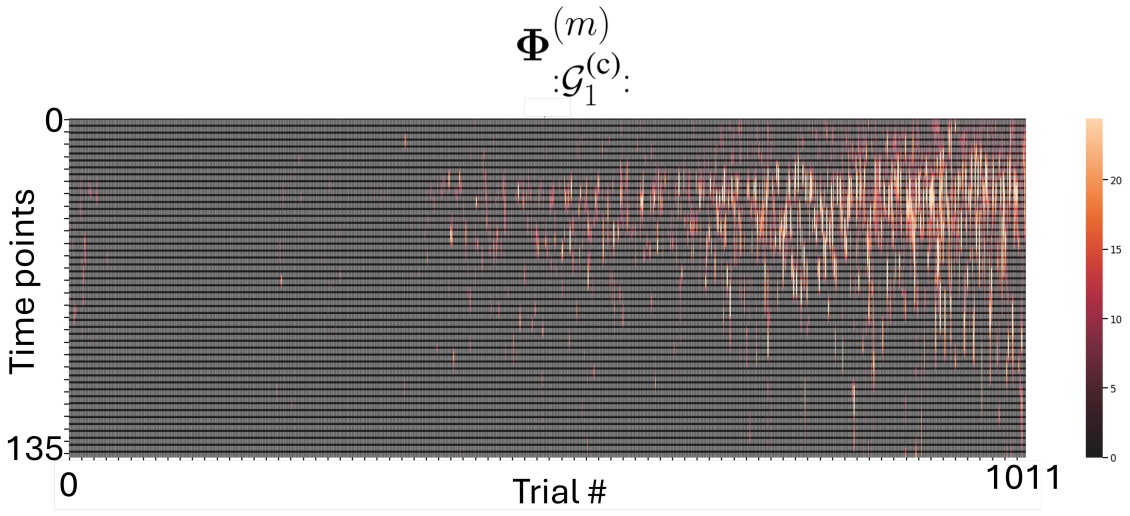

Figure 23: $\mathbf{\Phi}_1^{(c)}$ presents an increasing temporal drift over trials.

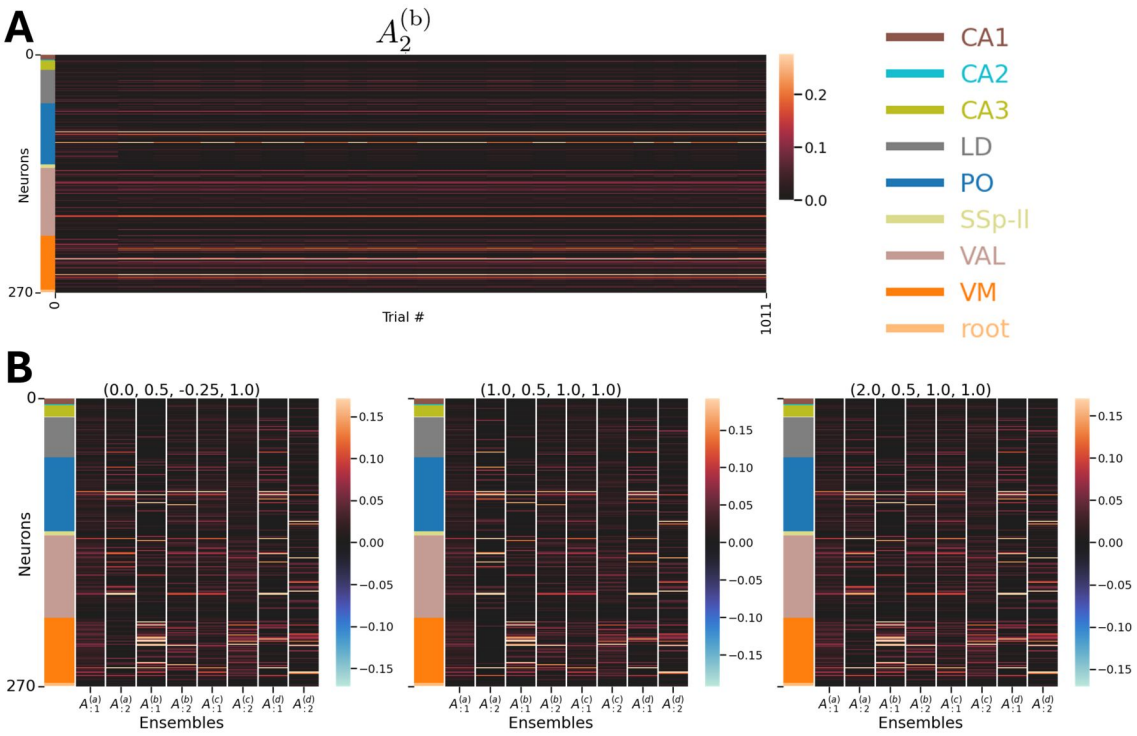

Figure 24: **Components identified by MILCCI in the IBL experiment. A:** Example ensemble (with its trace discussed in the main text) and its adjustments across trials. **B:** Example ensemble matrices reconstructed for three random trials. Each subplot shows all ensembles present in that trial under the unique set of labels indicated.

Abbreviation	State	Abbreviation	State
AL	ALABAMA	AK	ALASKA
AZ	ARIZONA	AR	ARKANSAS
CA	CALIFORNIA	CO	COLORADO
CT	CONNECTICUT	DE	DELAWARE
DC	DISTRICT OF COLUMBIA	FL	FLORIDA
GA	GEORGIA	HI	HAWAII
ID	IDAHO	IL	ILLINOIS
IN	INDIANA	IA	IOWA
KS	KANSAS	KY	KENTUCKY
LA	LOUISIANA	ME	MAINE
MD	MARYLAND	MA	MASSACHUSETTS
MI	MICHIGAN	MN	MINNESOTA
MS	MISSISSIPPI	MO	MISSOURI
MT	MONTANA	NE	NEBRASKA
NV	NEVADA	NH	NEW HAMPSHIRE
NJ	NEW JERSEY	NM	NEW MEXICO
NY	NEW YORK	NC	NORTH CAROLINA
ND	NORTH DAKOTA	OH	OHIO
OK	OKLAHOMA	OR	OREGON
PA	PENNSYLVANIA	RI	RHODE ISLAND
SC	SOUTH CAROLINA	SD	SOUTH DAKOTA
TN	TENNESSEE	TX	TEXAS
UT	UTAH	VT	VERMONT
VA	VIRGINIA	WA	WASHINGTON
WV	WEST VIRGINIA	WI	WISCONSIN
WY	WYOMING		

Table 1: List of US States and Their Abbreviations

normal perturbations with covariance scaled by σ^2, where $\sigma = 0.15$ ($\sigma^2 = 0.0225$), so trials that share a label exhibit correlated dynamics.

One component was designated as a random (trial-varying) component. Component activations were shifted to be nonnegative and rescaled so their 98th percentile matched the 98th percentile of the component maps. The observed data were produced by multiplying each trial's component-to-neuron map by that trial's temporal activations, yielding data of shape **(neurons $\times$ time $\times$ trials) = (80 $\times$ 500 $\times$ 250)**.

D Additional Information—Voting Experiment

D.1 Voting Data Pre-Processing

Data were acquired from Data & Lab (2017a;b;c), which included vote information for presidential, senate, and house elections in 51 states, including Washington, DC. The datasets cover the years 1976 to 2020 for presidential and senate elections, and 1976 to 2022 for house elections. For our analysis, we used the range 1976 to 2020 to ensure that all office types were included, resulting in 23 time points for house and senate, and 12 time points for presidential elections. For each year and each state, we took the total number of votes received by each party category (democrat, libertarian, republican, other) and divided each by the total number of votes cast in that state for that year's election. We excluded special elections. We designed the

Table 2: Parties Detailed vs. Simplified Versions. 'Other' includes only parties with at least 10 instances over all years & states.

Democrat	Republican	Other	Libertarian
Democrat; Democratic-Farmer-Labor; Democratic-Nonpartisan League; Democratic-Npl; Democrat (Not Identified On Ballot)	Republican	Prohibition; Independent; American Independent; U.S. Labor; Socialist Workers; American; Conservative; Socialist Labor; Independent American; Constitution; Socialist; Liberty Union; Statesman; Citizens; New Alliance; Workers World; Workers League; Independence; Populist; Nominated By Petition; Grassroots; No Party Affiliation; Green; Natural Law; Unaffiliated; Other; Working Families; Alliance; Non-Affiliated; Constitution Party; American Independent Party; Communist Party Use; Peace & Freedom; Taxpayers Party; Reform Party; U.S. Taxpayers Party; Socialism And Liberation Party; American Delta Party; American Solidarity Party; Party For Socialism And Liberation; Becoming One Nation	Libertarian

model to capture two following categories: 1) party (4 options), and 2) office (3 options). We ran MILCCI with $p^{(k)} = 4$ components for each category, that can structurally adjust per class, resulting in $p = 8$ total components. Due to the positive-only nature of the data, we applied a non-negativity constraint on both the unit-to-component memberships and the temporal traces.

D.2 Additional Findings—Voting Data

Some more interesting voting patterns, beyond these discussed in the main text, can be found in Figs. 12 and 8.

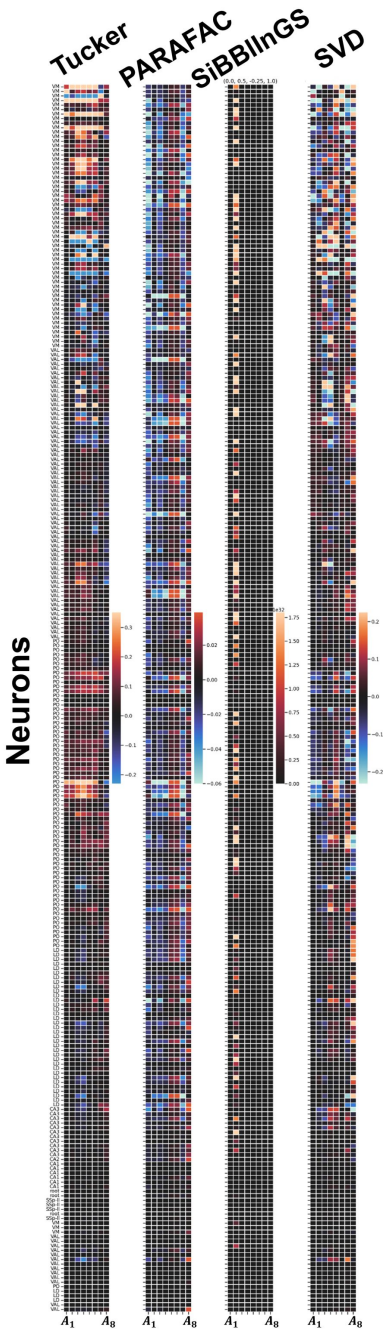

Figure 25: **IBL Neuronal Ensembles Components Identified by Baselines.**

In component $\boldsymbol{A}_4^{(\text{party})}$, Washington DC appears as its own distinct cluster. This likely reflects its highly unique political profile as the nation's capital, overwhelmingly Democratic, with very high voter turnout, which differs significantly from other states. In $\boldsymbol{A}_2^{(\text{office})}$ (i.e., the 6-th component overall), which unlike

previous components changes its composition depending on the office, West Virginia (WV) is included for House and Senate voting but excluded for the Presidency. This component reflects strong Democratic dominance in those legislative elections. The distinction for WV may align with its historical voting pattern of supporting Democrats more in local and state-level offices (House and Senate), while trending more Republican in presidential elections, reflecting a split in voter behavior based on the office. When exploring all traces colored by party (Fig. 12 top), Democrat vs. Republican is the dominant axis, consistently driving the most prominent distinctions over time. In some traces ($\mathbf{\Phi}_{:\mathcal{G}_1^{(\text{party})}}$, $\mathbf{\Phi}_{:\mathcal{G}_2^{(\text{party})}}$, $\mathbf{\Phi}_{:\mathcal{G}_3^{(\text{party})}}$), a noticeable divergence emerges around the year 2000, suggesting that the two parties began to separate more sharply around that period. This trend may correspond to the aftermath of the 2000 presidential election dispute (Bush vs Gore), which catalyzed a lasting partisan split, possibly further intensified by the ideological polarization following 9/11. Other traces, such as $\mathbf{\Phi}_{:\mathcal{G}_2^{(office)}:}$ and $\mathbf{\Phi}_{:\mathcal{G}_4^{(office)}}$, show broader fluctuations over time in opposite directions, which may reflect deeper, long-standing historical or structural differences between the parties. Trace $\mathbf{\Phi}_{:\mathcal{G}_3^{(office)}}$ appears to capture short-term variation or noise, including year-to-year peaks in party voting behavior. In contrast, the projections based on the office (Fig. 12) exhibit far fewer separations. The traces overlap substantially and show wide confidence intervals, suggesting that electoral behavior is less differentiated by office type than by party affiliation.

E Additional Information–Wikipedia Experiment

E.1 Wikipedia PageView Data Pre-Processing

We extracted daily Wikipedia pageview data from October 9, 2020, to October 29, 2024 ($T = 1482$ time points) for 48 diverse pages ("terms") related to college, computer science, machine learning, and psychology majors (Meta (2022)). Notably, we intentionally chose topics with both corollaries and co-variates. For each term, we collected data separately for three access platforms: (1) desktop, (2) mobile web, and (3) mobile app. We also distinguished the agent accessing the data: 1) a user or 2) a spider (for spider data was extracted only for web and desktop platform due to extreme sparsity of app + spider combination).

We focused on seven languages representing diverse world regions: English (en), Chinese (zh), Spanish (es), Hindi (hi), Arabic (ar), French (fr), and Hebrew (he). To ensure comparability, data for each language were range-normalized across all terms and time points using the 99th percentile to reduce outlier influence: $Y_{:,:,l} \leftarrow (Y_{:,:,l} - \min(Y_{:,:,l}))/\text{perc}(Y_{:,:,l}, 99)$, where $Y_{:,:,l}$ is the full dataset for language l.

Each term was then normalized across all languages and time points using the same 99th percentile procedure: $Y_{k,:,:} \leftarrow (Y_{k,:,:} - \min(Y_{k,:,:}))/\text{perc}(Y_{k,:,:}, 99)$, where $Y_{k,:,:}$ denotes the full time-course of term k across languages. This concludes to overall three categories candidate for compositional adjustments in the data: (1) agent (user or spider), (2) platform (desktop / mobile web / mobile app), and (3) language (one of the seven listed).

E.2 Clarification on Findings—Wikipedia Data

Lists of terms of components mentioned in main text (Sec. 4):

- $A_1^{(\text{agent})}$ **(Psychology)**: Classical conditioning; Bobo doll experiment; Operant conditioning; Self-concept; Little Albert experiment; Unsupervised learning; Embedding;
- $\boldsymbol{A}_2^{(\text{agent})}$**: (Social Media)**: The Social Network (movie); Social media; Ivan Pavlov;Mark Zuckerberg.
- $\boldsymbol{A}_4^{(\text{platform})}$**: (Computer Science)**: Data mining; Computer science; Supervised learning; Unsupervised learning; Computer scientist; Social media;

F Additional Information about Neuronal Ensembles Experiment

The IBL dataset is part of the International Brain Laboratory (IBL) effort to map neural activity underlying decision-making in mice across the whole brain. We accessed the IBL's data via the Dandi archive, in an NWB Rübel et al. (2022); Laboratory et al. (2025) format.

The randomly selected session was recorded on February 11, 2020 in the Churchland Lab at CSHL (currently at UCLA). In the IBL task, mice view a grating stimulus on the left or right side of a screen (or no stimulus) and report its location by turning a wheel. The task includes block-wise priors, where one side is more likely than the other, requiring mice to combine sensory evidence with prior expectations. Stimulus contrast varies across trials to manipulate difficulty, enabling precise measurement of perceptual decision-making and neural correlates during electrophysiology recordings. Electrophysiological recordings were collected using Neuropixels probes from diverse brain areas. These recordings provide single-spike resolution activity during a decision-making task and include additional data such as sensory stimuli presented to the mouse, behavioral responses and response times. The subject (ID: CSHL052) was a female C57BL/6 mouse (Mus musculus), 6 months old and 22g at the time of recording. The full description of the session and task protocol is provided in (Angelaki et al., 2025).

G Information About Baseline Calculation and Execution

We compared MILCCI to matrix (SVD), tensor (PARAFAC, HOSVD), and multi-array (SiBBlInGS (Mudrik et al., 2024)) decompositions.

G.1 Comparisons SVD, Tucker, PARAFAC

We compared these methods to MILCCI both quantitatively and qualitatively. For the qualitative comparison, we provide exemplary figures in the main text and appendix that emphasize MILCCI's superior performance. Notably, in all these methods the component matrix is fixed, as defined by the first mode; therefore, unlike MILCCI, they cannot (1) reveal structural adjustments over trials or disentangle category effects via the components, and (2) capture free trial-to-trial variability without tensor constraints (except for SVD). Consequently, their ability to capture such effects is inherently limited, though they represent the closest methods to MILCCI that we can reasonably compare to (in the sense that they provide comparable components and traces). Thus, the comparison is also limited in that we cannot show structural variability that these methods fundamentally do not support. Quantitatively, we calculated information criteria (AIC, BIC, HQC; lower values indicate better fit) using the degrees of freedom of each method. As seen in the appendix figures, these methods struggle to capture the data when constrained to the same dimensionality as MILCCI, which we attribute to (1) the need for small structural adjustments, and (2) their inability to capture free trial-to-trial variability.

See below running details for these methods:

- **SVD**: We used NumPy's 'linalg.svd' package, using the same number of components as in MILCCI for each experiment. The SVD was applied to the data from all trials concatenated horizontally. For experiments containing missing values (e.g., the voting experiment), NaNs were filled with zeros. Components were extracted from the left singular vectors (U), and the corresponding traces were obtained by multiplying the singular values matrix (Σ) with the right singular vectors ($V^\top$).
- **PARAFAC** (Harshman, 1970): We used the PyLops Ravasi & Vasconcelos (2020) PARAFAC implementation, with the same rank and number of components as MILCCI. For experiments with trials of varying durations (e.g., the IBL), we used the 90-th percentile trial length to prevent outliers from dominating, stacking trials along a third dimension and zero-padding shorter trials.

Components ($\boldsymbol{A}$) were extracted from the first tensor mode (first factor), and traces were obtained by multiplying the second mode and the third mode according to the trial and component count.

- **Tucker** (Tucker, 1966) (HOSVD): Also for Tucker, we used the PyLops Ravasi & Vasconcelos (2020) PARAFAC implementation, with the same dimensions and number of components as MILCCI. For the 3-rd mode, we used the minimum between the number of time points and the number of trials. Again, for experiments with trials of varying durations (e.g., the IBL), we used the 90-th percentile trial length to prevent outliers from dominating, stacking trials along a third dimension and zero-padding shorter trials.

  Components ($\boldsymbol{A}$) were extracted from the first tensor mode (first factor), and traces were obtained by multiplying the second mode, the core matrix, and the trial and component count. Notably, the component matrix in these methods is fixed, as defined by the core tensor, and therefore cannot adjust over time or disentangle label variability via the components.

For some datasets (e.g., Wikipedia), PARAFAC and Tucker (PyLops Ravasi & Vasconcelos (2020) implementation) could not converge at the same MILCCI dimensionality ($p = 12$), even with SVD initialization, various normalization schemes, high tolerance (1e-2), $\ell_2 2$ regularization, and maximum iterations of 10,000, due to least-squares optimization instability. We attribute this to: (1) many missing values (replaced with 0), (2) high-resolution (daily) measurements introducing considerable noise and temporal complexity, and (3) each trial being observed only once (i.e., a single observation per unique $L^{(m)}$). Hence, for the Wikipedia comparison, we used only SVD and SiBBlInGS (see below).

G.2 SiBBLInGS

:

MILCCI's (our method's) main advantage over SiBBlInGS is interpretability for multi-way, multi-label data. Particularly, SiBBlInGS cannot disentangle the effects of co- or separately-varying labels, making it difficult to understand their individual contributions. SiBBlInGS, by Modeling each unique label tuple as a distinct label would further increases tensor size and computational complexity. This is especially pronounced in experiments where some label categories vary across many unique values (e.g., IBL trial number with over 1000 values), or are continuous or non-continuous ordinal. MILCCI can handle all of these without requiring additional dimensions, and can also account for the ordinal nature of each category. In contrast, SiBBlInGS requires choosing a single sorting across all categories. Since SiBBlInGS applies multiple regularizations across axes, which hinders degrees-of-freedom calculation for information criteria, information-criteria comparison is intractable. Hence, we focused on qualitative comparisons that emphasize MILCCI's interpretability advantages. Notably, for all figures comparing SiBBlInGS and MILCCI (throughout the appendix), SiBBlInGS components vary uninterpretably across labels, while MILCCI clearly disentangles these effects. For visualization only, MILCCI was run with repeated columns for the same category under exemplary random trials. Importantly, all comparisons used identical initialization and parameters for both methods.

H Ethics Statement and LLM Usage

Our work does not raise any ethical concerns. All real data used are publicly available. For synthetic data, we provide the code used for generation. Large language models were employed only at the word or sentence level during manuscript writing to improve the language, with no influence on the scientific content or analysis.