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 variables spanning several categories. For example, a trial in a neuroscience study may be linked to a value from category (a): task difficulty, and category (b): animal choice. A critical challenge in time-series analysis is 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 approach to 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. Often, such metadata spans 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 such multi-trial, multi-label (*'multi-way'*) data, interpreting how the observations vary across the label space is complicated by the data's high dimensionality and trial-to-trial variability.

A practical approach is to analyze the data in a lower-dimensional latent space (Ma & Zhu, 2013), where label-related structures become easier to interpret. In this space, the activity can be described by a small set of *components*—units that capture the dominant sources of variability across trials and labels. Existing dimensionality reduction methods for analyzing multi-way data often factorize a single, large tensor into such components (Harshman, 1970), but they typically overlook trial labels and require constraints on the data structure (e.g., equal-length trials). An alternative is to apply factorizations separately to each trial, which can accommodate varying trial structure but sacrifices information about cross-trial relationships.

Hence, there is a need for new flexible yet interpretable methods to (1) discover 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 observations.

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 discovers interpretable sparse components underlying multi-way data and reveals how they capture diverse label categories.
- We identify components that capture label-driven variability across trials and track how their activations evolve within individual trials, thereby encompassing the full spectrum of trial-by-trial variability.
- We validate MILCCI on synthetic data, showing it better recovers true components than other methods.
- We 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. SliceTCA (Pellegrino et al., 2024) extends tensor factorization by simultaneously demixing neural, trial, and temporal covariability classes within the same dataset, allowing components to capture structure across different types of neural variability. However, sliceTCA does not incorporate explicit label information, is limited to tensor structure of the data, and does not inherently promote sparse components that *subtly* adjust compositions with corresponding freely varying traces (see Sec. H.2). 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 that vary over sessions, (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.

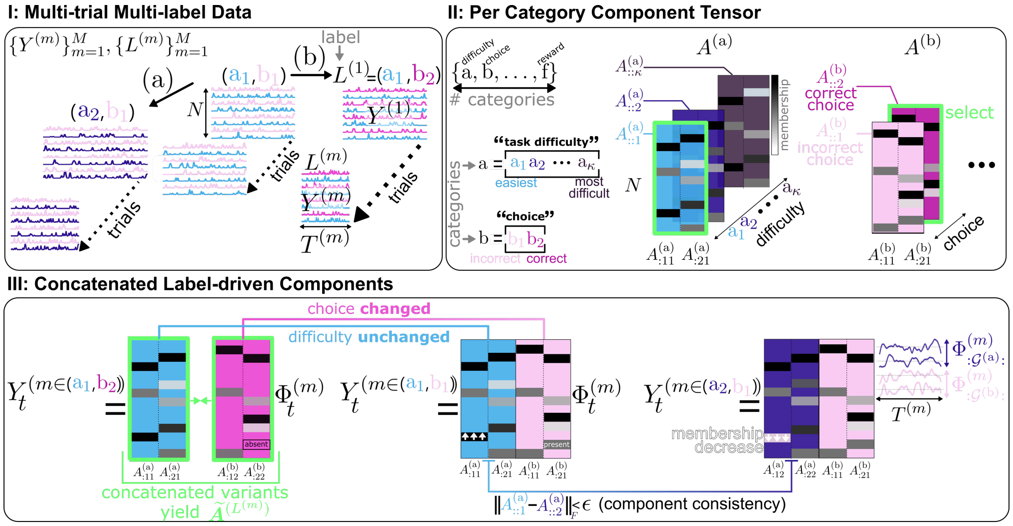

I: Multi-trial Multi-label Data
$\{Y^{(m)}\}_{m=1}^{M}, \{L^{(m)}\}_{m=1}^{M}$
label
(a)
(b)
$L^{(1)}$=(a$_1$,b$_2$)
$Y^{(1)}$
(a$_1$,b$_1$)
(a$_2$,b$_1$)
N
$L^{(m)}$
$Y^{(m)}$
$T^{(m)}$
trials
II: Per Category Component Tensor
difficulty
choice
reward
{a, b, . . . , f}
# categories
categories
"task difficulty"
a = a$_1$ a$_2$ ⋯ a$_\kappa$
easiest
most difficult
"choice"
b = b$_1$ b$_2$
incorrect
correct
$A^{(a)}$
$A^{(b)}$
$A^{(a)}_{::\kappa}$
$A^{(a)}_{::2}$
$A^{(a)}_{::1}$
membership
N
difficulty
$A^{(b)}_{::2}$ correct choice
$A^{(b)}_{::1}$ incorrect choice
select
choice
III: Concatenated Label-driven Components
choice changed
difficulty unchanged
$Y_t^{(m\in(a_1,b_2))}$ =
$\Phi_t^{(m)}$
$Y_t^{(m\in(a_1,b_1))}$ =
$Y_t^{(m\in(a_2,b_1))}$ =
absent
present
membership decrease
concatenated variants yield $\widetilde{A}^{(L^{(m)})}$
$\Phi^{(m)}_{:\mathcal{G}(a)}$
$\Phi^{(m)}_{:\mathcal{G}(b)}$
$T^{(m)}$
$\|A^{(a)}_{::1}-A^{(a)}_{::2}\|_F<\epsilon$ (component consistency)

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., $L^{(m)}$ = (easy task, correct choice)). **II:** Each category (k)'s components are represented by a tensor $\mathbf{A}^{(\mathrm{k})}$, whose i-th variant ($\mathbf{A}^{(\mathrm{k})}_{::i}$) refers to the i-th option of that category (e.g., if the 2-nd option of category (b): correct choice, then $\mathbf{A}^{(\mathrm{b})}_{::2}$ are correct-choice components). **III:** Each trial m is modeled via a sparse factorization, with its sparse components defined by selecting a variant (layer) from each category's tensor, based on that trial's label (green borders, **II**), and then concatenating all selected variants 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: $\|\boldsymbol{A}^{(\mathrm{a})}_{::1} - \boldsymbol{A}^{(\mathrm{a})}_{::2}\|_F < \epsilon$), and 3) component temporal traces ($\{\boldsymbol{\Phi}^{(m)}\}_{m=1}^{M}$) can vary flexibly across trials.

3 Our Component-identification Approach

Problem:

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 identities of the N channels are fixed across trials, while trial durations, $T^{(m)}$, can vary. Each trial $\boldsymbol{Y}^{(m)}$ is observed under trial-related metadata variables belonging to categories $C := \{\mathrm{a}, \mathrm{b}, \ldots, \mathrm{f}\}$ (e.g., (a) task difficulty, (b) animal's choice, etc.), such that $|C|$ is the number of categories. We define the *label* of trial m as the tuple $L^{(m)}$ containing its metadata variables (e.g., $L^{(m)}$ = (task difficulty: easy, choice: correct)), such that each *label entry* $L_i^{(m)}$ is the value of the i-th category in that trial. Notably, different trials may exhibit identical labels, partially overlapping labels, or entirely distinct labels. We aim to understand how these labels are encoded within the multi-way observations via their underlying components and their traces.

A parsimonious modeling strategy is to model observations in each trial m ($\boldsymbol{Y}^{(m)}$) as being linearly generated by a small set of P basis components that present temporally-evolving activations $\boldsymbol{\Phi}^{(m)} \in \mathbb{R}^{P\times T^{(m)}}$, such that $\boldsymbol{Y}^{(m)} = \boldsymbol{A}\boldsymbol{\Phi}^{(m)} + \epsilon$, with ϵ representing e.g., *i.i.d.* Gaussian noise. Such an approach, common in traditional matrix- or tensor-factorization, however, assumes that component compositions ($\boldsymbol{A}$) either vary freely across trials (if applied separately per trial) or remain fixed across all trials (if applied on a stacked version of the data). Consequently, they cannot capture cases where components *slightly* adjust their composition under label changes. For example, consider a brain network (i.e., a neuronal 'component') that recruits a few more neurons during a hard task absent from the network during an easy task. Methods based on linear scaling of fixed components, if applied independently to trials from each task difficulty, would fail to recognize the network as identical across these conditions. Alternatively, if applied to all trials stacked, they would force identical component structures for both easy and difficult tasks, which would misrepresent the new recruited neurons (e.g., by averaging their membership across the two task difficulties). Moreover, in such a case, as $\boldsymbol{Y} \approx \boldsymbol{A\Phi}$, inaccurate estimates of $\boldsymbol{A}$ would distort the estimation of $\boldsymbol{\Phi}$, which will compensate to maintain good data representations. Hence, there is a need for methods that can identify how labels are encoded, both structurally and temporally, within multi-way multi-label data.

Table 1: *Key notation used throughout the paper.*

Notation	**Description**
m	Trial #
(k)$\in \lvert C\rvert$	Category
$L^{(m)} = (L_1^{(m)}, \dots, L_{\lvert C\rvert}^{(m)})$	Label of trial m
$\boldsymbol{Y}^{(m)} \in \mathbb{R}^{N \times T^{(m)}}$	Trial m observations
$\boldsymbol{\Phi}^{(m)} \in \mathbb{R}^{P \times T^{(m)}}$	Trial m traces
$\mathcal{G}^{(\mathrm{k})}$	Indices of category (k) traces for cat (k)
$\boldsymbol{A}^{(\mathrm{k})} \in \mathbb{R}^{N \times p^{(\mathrm{k})} \times \lvert\mathrm{k}\rvert}$	Category (k) components
$L^{(m)} = (L_1^{(m)}, \dots, L_{\lvert C\rvert}^{(m)})$	Label of trial m
$\widetilde{\boldsymbol{A}}^{(L^{(m)})} \in \mathbb{R}^{N \times P}$	Trial m loading

Our Model:

MILCCI assumes that each trial's observations arise from the decomposition of $|C| \in \mathbb{Z}_{>0}$ matrices of components, $\{\boldsymbol{A}^{(\mathrm{k})}\}_{(\mathrm{k}) \in C}$, where each matrix $\boldsymbol{A}^{(\mathrm{k})} \in \mathbb{R}^{N \times p^{(\mathrm{k})}}$ represents the components for category (k) $\in C$. Each entry $\boldsymbol{A}^{(\mathrm{k})}_{nj}$ captures channel n's membership degree in the j-th component (e.g., the extent to which neuron n participates in a neuronal ensemble), where $\boldsymbol{A}^{(\mathrm{k})}_{nj} = 0$ indicates non-membership (Fig 1, **II**). We assume that component memberships are sparse, i.e., each channel n belongs to *only a few* components. Hence, we place a Laplace prior on each entry: $\boldsymbol{A}^{(\mathrm{k})}_{nj} \sim \text{Laplace}(0, \frac{1}{\gamma_1})$, where γ_1 is a sparsity-scaling parameter (see Sec. C for details).

Each of the P components also exhibits a time-varying trace within each trial, collectively represented by the rows of the per-trial traces matrix $\boldsymbol{\Phi}^{(m)} \in \mathbb{R}^{P \times T} \sim \mathcal{MN}_{P \times T}(\boldsymbol{0}, \sigma^2 \mathbf{I}_P, \mathbf{I}_T)$, where σ is a scalar that controls the prior's variance and $\mathbf{I}_P$ and $\mathbf{I}_T$ are identity matrices of dimensions P and T. In each trial $m = 1, \dots, M$ with associated label $L^{(m)}$, we aim to model the observations $\boldsymbol{Y}^{(m)}$ as a decomposition of the label-driven components weighted by their corresponding traces: $\boldsymbol{Y}^{(m)} \approx \sum_{(\mathrm{k}) \in C} \boldsymbol{A}^{(\mathrm{k})} \boldsymbol{\Phi}^{(m)}_{\mathcal{G}^{(\mathrm{k})},:}$, where $\mathcal{G}_{\mathrm{k}}$ are the row indices of the traces of category (k)'s components.

Here, however, we depart from the fixed-component assumption underlying the decomposition above, and instead assume that components can exhibit compositional variants by subtly adjusting their memberships under label changes. Formally, we posit that each $\boldsymbol{A}^{(\mathrm{k})}$ comprises $|\mathrm{k}|$ variants (Fig. 1, **II**), where $|\mathrm{k}|$ equals the number of unique label options for that category (e.g., $\boldsymbol{A}^{(\text{choice})}$ has $|\mathrm{k}| = 2$ variants for binary choice tasks). Hence, we generalize the above decomposition by extending each component matrix to a 3D tensor $\boldsymbol{A}^{(\mathrm{k})} \in \mathbb{R}^{N \times p^{(\mathrm{k})} \times |\mathrm{k}|}$, such that $\boldsymbol{A}^{(\mathrm{k})}_{::i}$ is the i-th variant of $\boldsymbol{A}^{(\mathrm{k})}$. Importantly, different tensors in $\{\boldsymbol{A}^{(\mathrm{k})}\}_{\mathrm{k} \in C}$ can be associated with distinct numbers of variants (and thus have varying 3rd dimensions), depending on how many unique values their corresponding category contains (Fig. 1, II: κ options for task difficulty vs. 2 options for choice).

While the variants of each component need not be structurally identical, our model assumes they remain close to each other in a way that reflects the similarity between their label values (for distinct options $\mathrm{k}_i \neq \mathrm{k}_{i'}$ within category k, $\|\boldsymbol{A}^{(\mathrm{k})}_{::i} - \boldsymbol{A}^{(\mathrm{k})}_{::i'}\|_F^2 < \delta(\mathrm{k}_i, \mathrm{k}_{i'})$ for small, category-dependent $\delta(\mathrm{k}_i, \mathrm{k}_{i'}) \in \mathbb{R}$) to preserve component identity. Importantly, as the number of components P is limited across trials, the $\{\boldsymbol{A}^{(\mathrm{k})}\}$ tensors must learn meaningful synergies between different label combinations rather than provide simple one-to-one per-trial mappings. This constraint promotes alignment of components across trials and enables revealing cross-label shared underlying patterns that would otherwise remain obscured.

The decomposition model is thereby extended to the following label-specific formulation for each trial m:

$$\boldsymbol{Y}^{(m)} = \sum_{(\mathrm{k})\in C} \boldsymbol{A}^{(\mathrm{k})}_{::L^{(m)}_{\mathrm{k}}} \boldsymbol{\Phi}^{(m)}_{\mathcal{G}^{(\mathrm{k})}:} + \boldsymbol{\epsilon}, \quad \epsilon_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2) \tag{1}$$

Model Fitting Procedure:

Learning the component compositions and their traces directly from $\{\boldsymbol{Y}^{(m)}\}$ is hindered by the mixing of all labels' effects across categories. We address this through a three-stage procedure:

- *Stage 1: Pre-computing label similarity graphs.* To accommodate both categorical and ordinal labels, MILCCI includes a pre-computing step that calculates a label similarity graph $\boldsymbol{\lambda}^{(\mathrm{k})} \in \mathbb{R}^{|\mathrm{k}|\times|\mathrm{k}|}$ for each category k (see graph calculation details in App. A.2). These graphs can be integrated into *Stage 3* to ensure that compositional adjustments reflect label-to-label distances for ordinal labels.
- *Stage 2: Initialization.* We initialize the components and traces following Appendix A.1.
- *Stage 3: Iterative optimization.* Updating $\{\boldsymbol{A}^{(\mathrm{k})}\}_{\mathrm{k}\in\mathrm{C}}$, $\{\boldsymbol{\Phi}^{(m)}\}_{m=1}^{M}$ until convergence as detailed below.

Inferring $\{\boldsymbol{A}^{(\mathbf{k})}\}_{\mathbf{k}\in\mathbf{C}}$: For each unique category (k)'s option k_i (e.g., each task difficulty level), we infer the variant $\boldsymbol{A}^{(\mathrm{k})}_{::i}$ using all trials observed under k_i ($\widetilde{M} = \{m \in \{1, \dots, M\} \mid L^{(m)}_{\mathrm{k}} = \mathrm{k}_i\}$), regardless of those trials' values in other categories (e.g., all trials where difficulty: easy, regardless of choice).

Since each trial m is modeled via a decomposition of components from multiple categories (Eq. 1), to learn this variant's ($\boldsymbol{A}^{(\mathrm{k})}_{::i}$) structure, we need to separate its per-trial effect from those of the other categories' components. Hence, for each $m \in \widetilde{M}$, we first calculate the residual matrix $\widetilde{\boldsymbol{Y}}^{(m,\mathrm{k})}$, which represents the difference between $\boldsymbol{Y}^{(m)}$ and its partial reconstruction based on all components excluding $\boldsymbol{A}^{(\mathrm{k})}_{::i}$, specifically: $\widetilde{\boldsymbol{Y}}^{(m,\mathrm{k})} := \boldsymbol{Y}^{(m)} - \sum_{\mathrm{k'}\neq\mathrm{k}} \boldsymbol{A}^{(\mathrm{k'})}_{::L^{(m)}_{\mathrm{k'}}} \boldsymbol{\Phi}^{(m)}_{\mathcal{G}^{(\mathrm{k'})}:}$. We then infer $\boldsymbol{A}^{(\mathrm{k})}_{::i}$ via LASSO (Tibshirani, 1996):

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

where $\boldsymbol{A}^{(\mathrm{k})}_{::i} \geq 0$ applies an optional non-negativity constraint on the components, $\lambda^{(\mathrm{k})}_{i',i}$ promotes similarity among same-category (k) variants (App. A.2), and γ_2, γ_1 are hyperparameters for cross-label smoothness and sparsity. Collectively, Equation 2 balances data fidelity (1st term), consistency between corresponding components proportional to their label similarity by $\boldsymbol{\lambda}^{(\mathrm{k})}$ (2nd), and component sparsity (3rd). Each component is then normalized to a fixed sum to avoid scaling ambiguity with $\boldsymbol{\Phi}^{(m)}$.

Updating $\{\boldsymbol{\Phi}^{(\mathbf{m})}\}_{m=1}^{M}$: Since traces vary independently of labels across trials (i.e., unsupervised), they can be learned per trial m using the label-driven loading matrix of realized components in that trial. This loading matrix $\widetilde{\boldsymbol{A}}^{(m)} := [\boldsymbol{A}^{(\mathrm{a})}_{::L^{(m)}_1}, \boldsymbol{A}^{(\mathrm{b})}_{::L^{(m)}_2}, \dots] \in \mathbb{R}^{N\times P}$ is constructed by horizontal concatenation of the appropriate variants from all categories as defined by the trial's label $L^{(m)}$ (Fig. 1, **III**). For each trial $m = 1 \dots M$, $\boldsymbol{\Phi}^{(m)}$ is then updated by:

$$\widehat{\boldsymbol{\Phi}}^{(m)} = \arg \min_{\boldsymbol{\Phi}^{(m)}} \underbrace{\|\boldsymbol{Y}^{(\mathrm{m})} - \widetilde{\boldsymbol{A}}^{(L^{(m)})}\boldsymbol{\Phi}^{(m)}\|_F^2}_{\text{data fidelity}} + \gamma_3 \underbrace{\sum_{t=1}^{T^{(m)}} \|\boldsymbol{\Phi}^{(m)}_{:,t} - \boldsymbol{\Phi}^{(m)}_{:,t-1}\|_2^2}_{\text{temporal smoothness}} + \gamma_4 \underbrace{\|(\mathbf{C} \odot (\mathbf{1} - \mathbf{I}_P)) \odot \boldsymbol{D}\|_{1,1}}_{\text{within-trial trace decorrelation}}, \tag{3}$$

where γ_3, γ_4 are hyperparameters, $\odot$ is element-wise multiplication, $\mathbf{C} := \mathrm{Gram}(\boldsymbol{\Phi}^{(m)})$, and $\boldsymbol{D} \in \mathbb{R}^{p\times p}$ is for normalization ($D_{j,j'} := \|\boldsymbol{\Phi}^{(m)}_{:j}\|_2^{-1}\|\boldsymbol{\Phi}^{(m)}_{:j'}\|_2^{-1}$). Eq. 3 overall promotes data fidelity (1st term), smoothness (2nd), and trace decorrelations (3rd). See algorithm, notations, and illustration in Alg.1, Tab.1, and Fig. 1.

4 Experiments

We validate MILCCI on synthetic data and demonstrate its components on four real-world datasets.

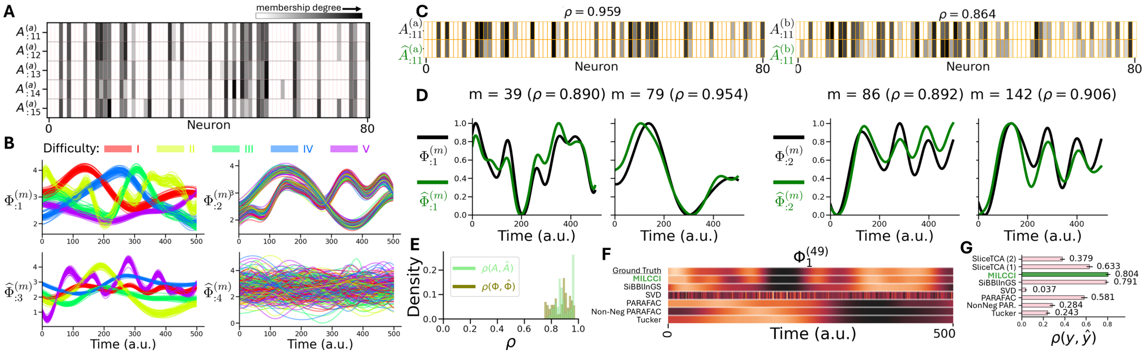


Figure 2: **MILCCI Recovers True Representations in Synthetic Data. A&B**: Generated synthetic data (full data in Fig. 6). Ground-truth components (examples in panel **A**) vary slightly across labels but remain fixed across same-label trials (rows). Ground-truth traces vary across trials (**B**, colored 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 reconstruction performance (**G**, baselines details in Sec. H).

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 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. 6B). 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. 6, A, App. D). We ran MILCCI on this data, comparing to (1) Tucker (Tucker, 1966), (2) PARAFAC (Harshman, 1970), (3) non-negative PARAFAC, (4) SVD, (5) SiBBlInGS (Mudrik et al., 2024), and (6) sliceTCA (Pellegrino et al., 2024); all using the same $P = 4$ components (Fig. 2, F,G, Sec.H). Since TF is invariant to component permutation, we used linear sum assignment to align the components of each method 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 blurred traces compared to the ground truth. 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: We next tested MILCCI on voting data consist of 50 US states × 23 years (sampling every 2 or 4 years) × party (e.g., Democrat, Republican, Libertarian; see Tab. 3) × office (Presidency, Senate, House) that we extracted from (Data & Lab, 2017a;b;c) and normalized for each state (App. E). Some states lack data for certain years or office due to differing election schedules (Fig. 7).

We applied MILCCI 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. 15, right) and the Constitution Party (PA, Fig. 15, 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

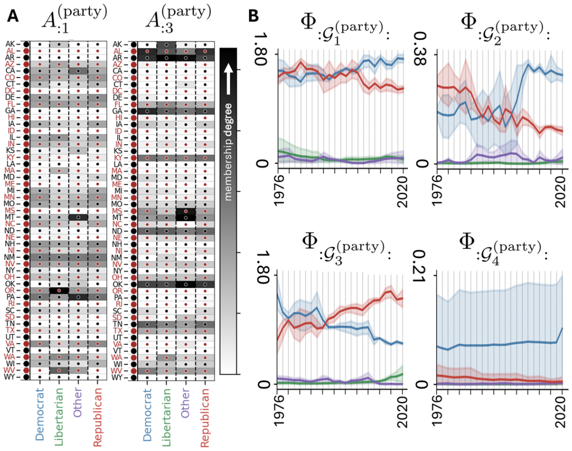


Figure 3: **Voting Results:** Identified example components (**A**) and traces (**B**, Mean and 80% confidence interval). See full in Fig. 7.

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 ($\mathbf{\Phi}_{:\mathcal{G}^{(\text{party})}_1}$, Fig. 3, B.1, top-left) shows overlapping Democrat–Republican activations diverging ∼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 underlying trends directly from observations. Temporally, its trace $\mathbf{\Phi}_{:\mathcal{G}^{(\text{party})}_3}$ (Fig. 3, B, 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. E.2) demonstrate its ability to uncover state–party–office–dependent patterns. Notably, components from other TF methods (Fig. 12,9) are dense (PARAFAC), include negative values (SVD, PARAFAC, Tucker), or, in SiBBlInGS, fail to disentangle party from office, which hinder interpretability (Fig. 11). 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. F.1. We ran MILCCI on the data with $p_j = 4$ components per category, and identified components that cluster related Wiki-pages together and vary across categories (Fig. 4, 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. 4, A, red arrows, list in App. F.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. 4, A, 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

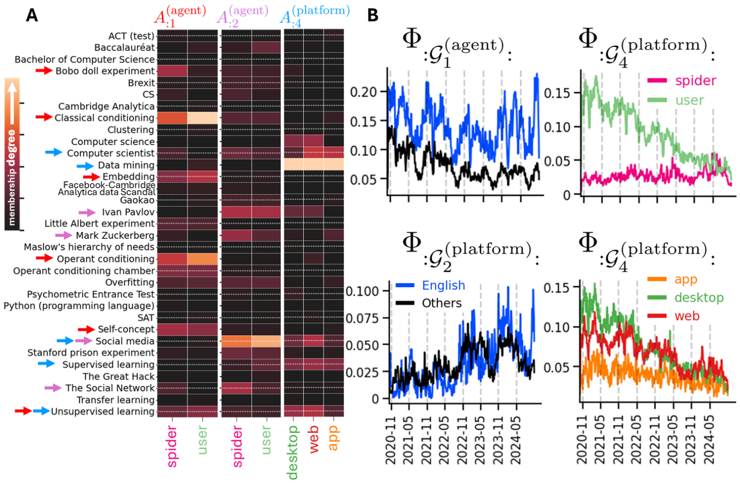


Figure 4: Identified Wiki page-view example components (**A**) and traces averaged by different categories (**B**).

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. 22). This component's trace ($\mathbf{\Phi}_{:\mathcal{G}_1^{(\text{agent})}:}$, Fig. 4, B top-left) captures its fluctuations and higher activation in English compared to other languages. This aligns with professional terms being more elaborate in English, often using jargon not fully defined/used in other languages, and with non-English native speakers possibly preferring to read professional material in English (Miquel-Ribé & Laniado, 2018).

The social-media component $\boldsymbol{A}_2^{(\text{agent})}$ 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 ($\mathbf{\Phi}_{:\mathcal{G}_2^{(\text{agent})}:}$) rises until a peak in Mar. 2024 in both English and non-English (Fig. 4 B, 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.

$\boldsymbol{A}_4^{(\text{platform})}$ (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 ($\mathbf{\Phi}_{:\mathcal{G}_4^{(\text{platform})}}$) shows higher activity in English (Fig. 17), with access dominated by desktop and mobile web (Fig. 4, B, bottom-right). The temporal trace is mostly driven by users rather than spiders (Fig. 4, B, 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. 17, 20, 19, 21, 22 for comparisons and full traces.

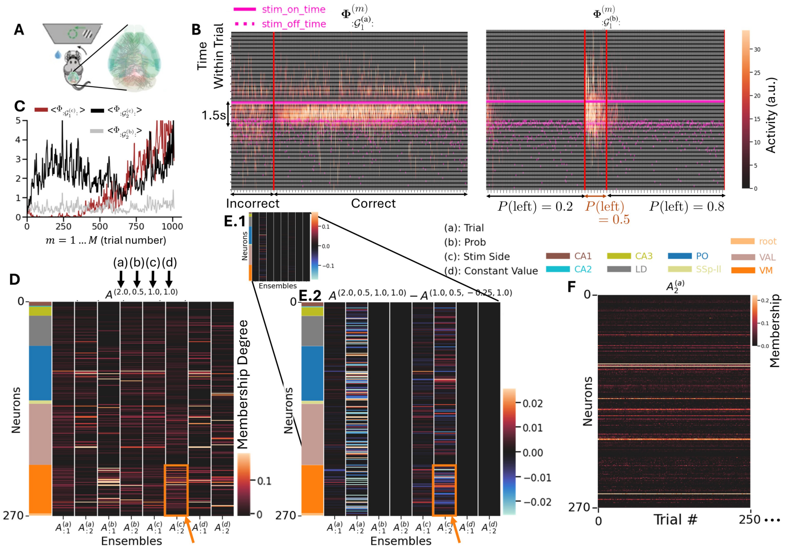


Figure 5: **MILCCI identifies meaningful neuronal ensembles in real-world brain data. A**: Experimental setting (from Angelaki et al. (2025)). **B**: Traces (all 1011 trials) sorted horizontally by trial correctness (left panel) and $p(\text{left})$ (right panel). **C**: Average within-trial values of exemplary traces reveal varying degrees of temporal drifts over trials. **D**: Ensembles identified (example trial). **E**: Differences in ensemble composition across trials. **F**: Trial-adjusted ensemble compositions over first 250 trials.

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. G). In this experiment mice reported the location of a visual grating with varying contrast by turning a wheel left or right (Fig. 5, 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. 30). 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. 5). For example, $\mathbf{\Phi}^{(m)}_{:\mathcal{G}_1^{(\text{a})}}$ is tuned to choice correctness, with activity surging during stimulus presentation (Fig. 5, B left) under *correct* choice only. Interestingly, MILCCI also found an ensemble sensitive to *incorrect* decisions, activated just after stimulus presentation ($\mathbf{\Phi}_{:\mathcal{G}_2^{(\text{b})}:}$, Fig. 31). 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. 33). In another example, $\mathbf{\Phi}^{(m)}_{:\mathcal{G}_1^{(\text{b})}:}$ 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. 5, B right). This highlights MILCCI’s efficiency in suggesting involvement of priors in decision-making, and its capability exposing similarities of similar-condition traces.

Moreover, MILCCI’s ability to isolate components with characteristics that track temporal drifts ($\mathbf{\Phi}^{(m)}_{:\mathcal{G}_1^{(\text{c})},:}$, Fig. 5, C brown; 32) 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. 5, D; colors on the left mark brain regions of neurons) capture distinct local structures. For example, $\boldsymbol{A}^{(c)}_{:2}$ captures many neurons in the Ventral Medial nucleus of the thalamus (VM, orange arrow in Fig. 5D), suggesting it may be involved in arousal regulation and motor coordination. We can also identify interesting patterns via the ensemble compositions (Fig. 5, D), and how they change between trials (e.g., between two example trials with differing stimulus sides; Fig. 5, E.1, zoom-in on changes in Fig. 5, E.2). These reveal patterns in ensemble composition adjustments: the first ensemble of $\boldsymbol{A}^{(a)}$ (Fig. 5, E.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 localization around VM (e.g., $\boldsymbol{A}^{(c)}_{:2}$, Fig. 5), suggesting an adaptive VM composition.

5 Discussion, Limitations, and Future Steps:

We presented MILCCI, a data-driven method for analyzing multi-trial, multi-label time series. MILCCI identifies interpretable components underlying the data, captures cross-trial relationships, and integrates label information, while accounting for trial-to-trial variability beyond label effects. MILCCI allows components to adjust their composition with label changes, which uncovers similarities that could remain hidden under fixed-component factorizations. Unlike other 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 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 alternatives, and demonstrated its effectiveness on real-world data: (1) exposing voting patterns aligned with real events; (2) recovering interpretable Wikipedia view components with memberships varying across languages, platforms, and agents; and (3) revealing neuronal ensembles adapting across trials and task variables. Notably, MILCCI's architecture (low-rank, sparse components) is robust to noise by constraining the model to learn patterns that are reused and generalize across trials under varying label combinations.Another MILCCI advantage is its applicability to diverse tensor decompositions beyond temporal data, though we focused on time series here for clarity.

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. This first MILCCI version assumes a linear decomposition within each trial for interpretability and consistency with other multi-way decompositions. Capturing nonlinear structures beyond preprocessing transformations (e.g., tanh applied pre-fitting, Appendix J.1) is an exciting future direction. This could be achieved through a kernelized version of MILCCI that operates in implicit feature spaces, or by directly modeling nonlinear relationships using gradient-based optimization methods. For large datasets, MILCCI, similarly to any iterative optimization method, may take longer, which can be addressed via parallel optimization or batch processing. MILCCI's modular design supports substitution of our current MSE metric (reflecting normal distribution assumptions) with alternative distributions or cost metrics, broadening its applicability in future extensions . 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); Mudrik et al. (2025)).

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.

Yanyuan Ma and Liping Zhu. A review on dimension reduction. *International Statistical Review*, 81(1): 134–150, 2013.

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.

Noga Mudrik, Ryan Ly, Oliver Ruebel, and Adam Shabti Charles. Creimbo: Cross-regional ensemble interactions in multi-view brain observations. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Lia Papadopoulos, Suhyun Jo, Kevin Zumwalt, Michael Wehr, David A McCormick, and Luca Mazzucato. Modulation of metastable ensemble dynamics explains optimal coding at moderate arousal in auditory cortex. *bioRxiv*, 2024.

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.

Arthur Pellegrino, Heike Stein, and N Alex Cayco-Gajic. Dimensionality reduction beyond neural subspaces with slice tensor component analysis. *Nature Neuroscience*, 27(6):1199–1210, 2024.

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 Additional Fitting 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 / incorrect, 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.

A.3 About sparsity and consistency hyperparameters

Like most machine learning models, MILCCI includes hyperparameters that control model behavior, though notably fewer than complex models such as deep networks. Two key hyperparameters in MILCCI control component behavior: γ_1 (sparsity) and γ_2 (cross-label consistency).

γ_1 controls the ℓ_1 regularization on component memberships (Eq. 2). Higher γ_1 values produce sparser components with fewer non-zero entries, which is crucial for interpretability but potentially missing weaker relationships. Lower γ_1 values allow denser components that capture more subtle patterns but may include noise. γ_2 promotes similarity between component variants within the same category via ℓ_2 regularization on their distances (Eq. 2). Higher γ_2 values force variants to remain nearly identical, losing the ability to capture label-driven adjustments. Lower γ_2 values allow more flexibility but risk components diverging too much across labels.

We recommend selecting γ_1 by examining component sparsity and interpretability (e.g., inspecting the number of non-zero entries and their meaningfulness), and by testing information criteria (e.g., AIC, BIC) post training. For γ_2, we recommend analyzing the distribution of pairwise distances between same-category component variants to ensure they remain similar yet allow meaningful adjustments. When domain knowledge is available (e.g., in neuroscience, we may have an estimate of how many neurons form a group based on the amount of data we recorded), this can further guide hyperparameter selection. In practice, we found MILCCI's performance relatively robust across reasonable hyperparameter ranges: we tested 20 values for $\gamma_1 \in [0.002, 0.5]$ and 20 values for $\gamma_2 \in [0.0002, 0.5]$ (800 total combinations, see Sec. J.3 and Fig. 37).

B Running Details

B.1 Data and Code Availability

All real-world data used in this paper are publicly available online, with sources cited in the corresponding sections. Code for the model implementation and synthetic data generation will be made available upon publication.

B.2 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.

C Elastic Net Prior of Components

In Section 3, we specified that the components follow a Laplace distribution:

$$\boldsymbol{A}^{(\text{k})}_{nj} \sim \text{Laplace}(0, \frac{1}{\gamma_1}) = \frac{\gamma_1}{2} \exp\left(-\gamma_1 |\boldsymbol{A}^{(\text{k})}_{nj}|\right)$$

When we later extended the model to multiple component variants for each category k with constrained ℓ_2 distances between them (i.e., $\|\boldsymbol{A}^{(\text{k})}_{n:i} - \boldsymbol{A}^{(\text{k})}_{n:i'}\|_2 < \epsilon$), we essentially employ a hierarchical Bayesian framework. The variant-specific components $\boldsymbol{A}^{(\text{k})}_{n:i}$ thus follow an elastic net prior:

$$p(\boldsymbol{A}) \propto \exp\left(-\gamma_1 \sum_{\text{k}} \sum_{n,j} |\boldsymbol{A}^{(\text{k})}_{nj}| - \gamma_2 \sum_{i' \neq i} \lambda^{(\text{k})}_{i',i} \|\boldsymbol{A}^{(\text{k})}_{::i'} - \boldsymbol{A}^{(\text{k})}_{::i}\|_2^2\right)$$

The first term corresponds to the ℓ_1 penalty inherited from the Laplace prior, promoting sparsity. The second term introduces an ℓ_2 penalty on variant differences, encouraging similarity within variant groups.

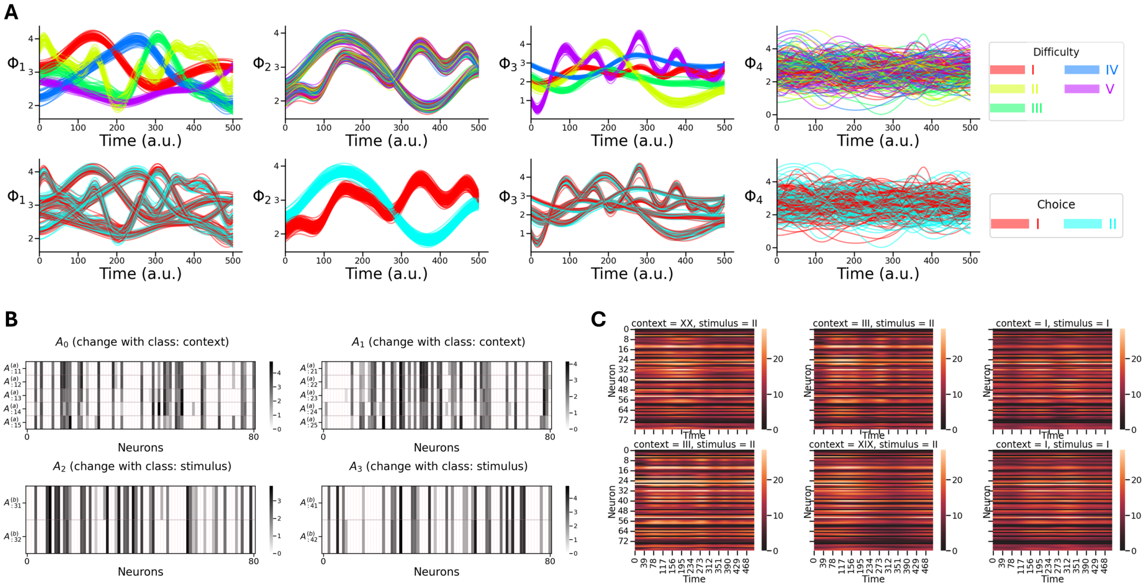


Figure 6: **Generated Synthetic Data. A:** Generated traces, colored by difficulty (top) or choice (bottom). **B:** Generated components. Each subplot shows one component and how it varies over the labels of each category (changes across rows). In other words, each subplot corresponds to $\boldsymbol{A}^{(\mathrm{k})}_{:j:}$ for some component j. **C:** Random example generated synthetic trials $\{\boldsymbol{Y}^{(m)}\}$.

This combination yields elastic net regularization, emerging naturally from the hierarchical structure where variants share statistical strength through their common base component.

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 $\mathrm{k}_i$ do
            Compute residuals for trials with label $\mathrm{k}_i$
            Solve for $\boldsymbol{A}^{(\mathrm{k})}_{::i}$ with cross-label consistency and sparsity via LASSO    ▷ equation 2
            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 $\Phi$
    for each trial $m$ with label $\ell$ do
        Build the stacked component matrix $\widetilde{\boldsymbol{A}}^{(\ell)}$ by selecting a variant from each $\boldsymbol{A}^{(\mathrm{k})}$.
        Update traces $\boldsymbol{\Phi}^{(m)}$ to minimize data fidelity, smoothness and de-correlation    ▷ equation 3
```

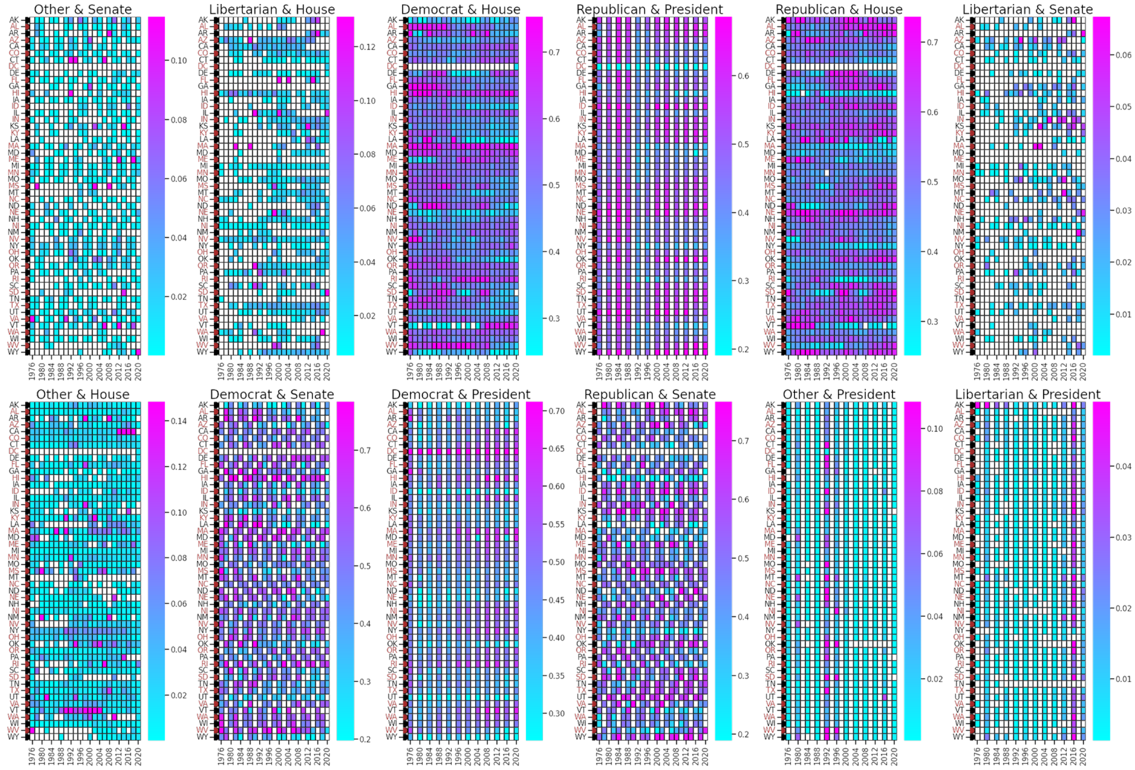


Figure 7: **Voting Data**: ($\{\boldsymbol{Y}^{(m)}\}_{m=1}^{M}$) Show Diverse Data Structures

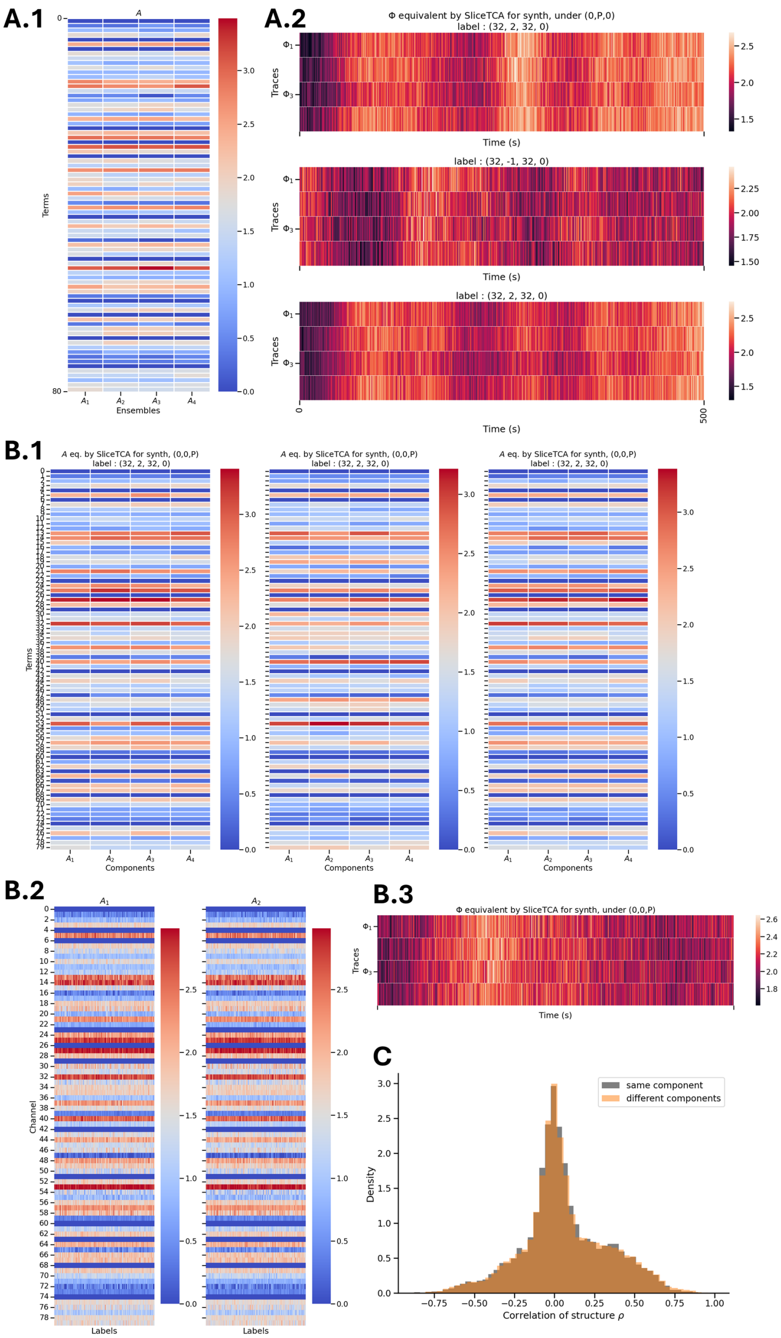
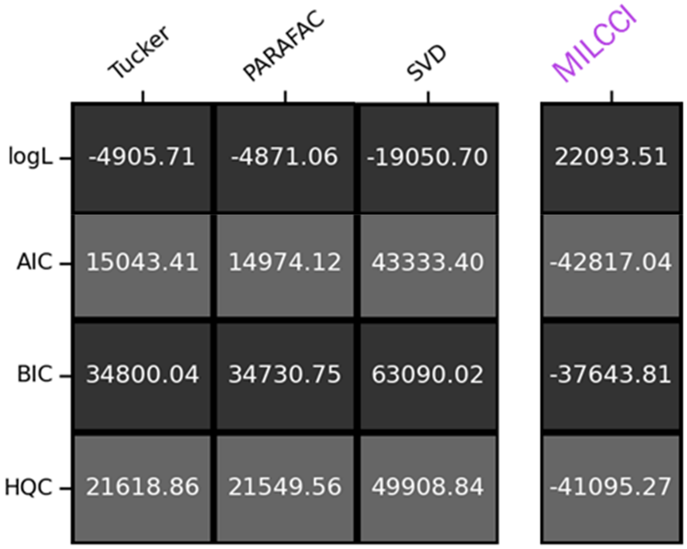


Figure 8: Components and Traces Identified by SliceTCA on synthetic data. **A** Results for configuration 1 (Sec. H.2), where **A.1** represents the components and **A.2** represents the corresponding traces from sliceTCA's $\boldsymbol{A}$ matrix. **B** Results for sliceTCA's configuration 2. **B.1** Identified components for 3 example trials; each subplot represents all components of one trial. **B.2** Shows how identified components vary over trials. **B.1 & B.2** together show that sliceTCA identifies components that are very close to each other, with cross-component variability similar to the variability of the same component over trials. This suggests that components are not necessarily matched over trials in terms of identity, as seen in panel **C** which shows the correlation distribution between same components and different components. **B.3** The temporal traces obtained by sliceTCA for configuration 2, shared across all trials.



	Tucker	PARAFAC	SVD	MILCCI
logL	-4905.71	-4871.06	-19050.70	22093.51
AIC	15043.41	14974.12	43333.40	-42817.04
BIC	34800.04	34730.75	63090.02	-37643.81
HQC	21618.86	21549.56	49908.84	-41095.27

Figure 9: **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.

D Additional Information—Synthetic Experiment

We generated synthetic datasets with **80 channels**, **4 components** (2 categories, × 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 $\{\text{I}, \text{II}, \text{III}, \text{VI}, \text{V}\}$ (5 levels) and category (b) (choice) labels from $\{\text{I}, \text{II}\}$ (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 variants 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-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** × **time** × **trials) = (80** × **500** × **250)**.

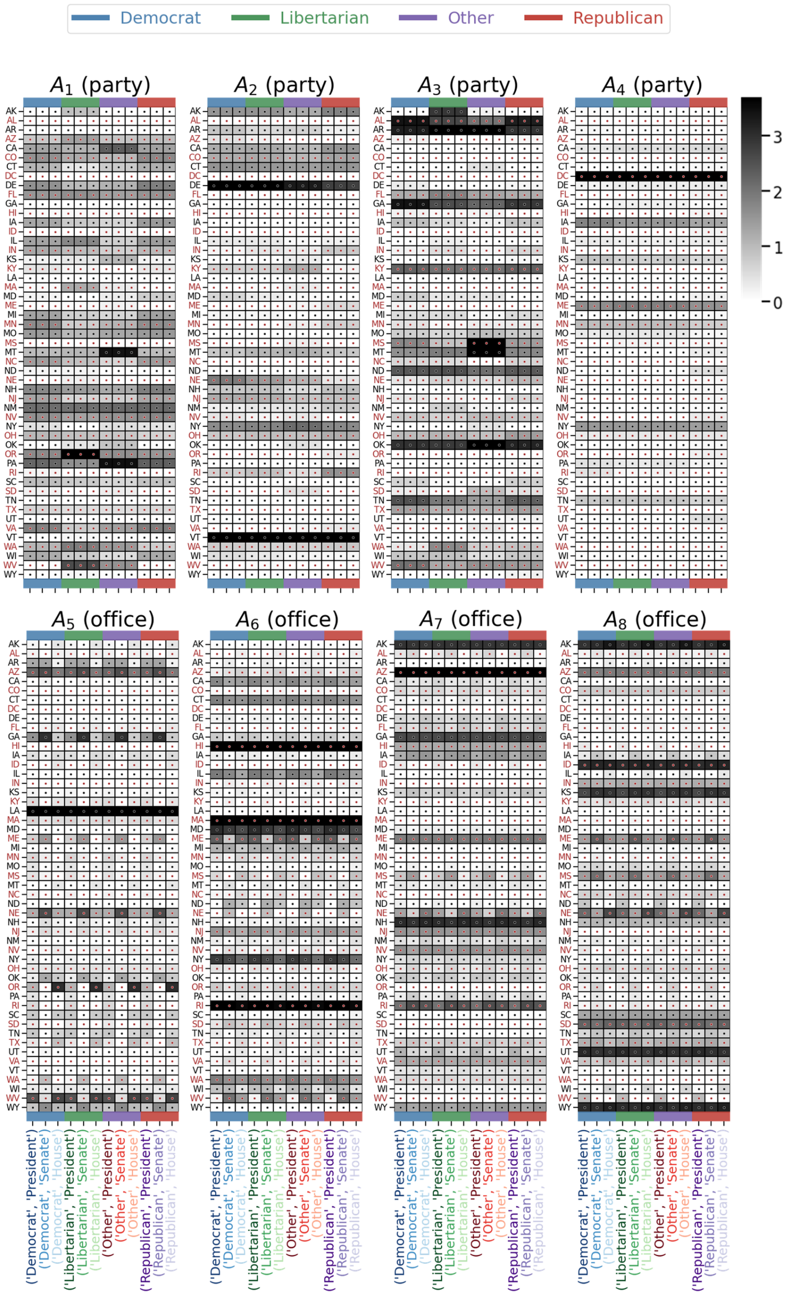


Figure 10: Identified Ensembles for Voting Experiment.

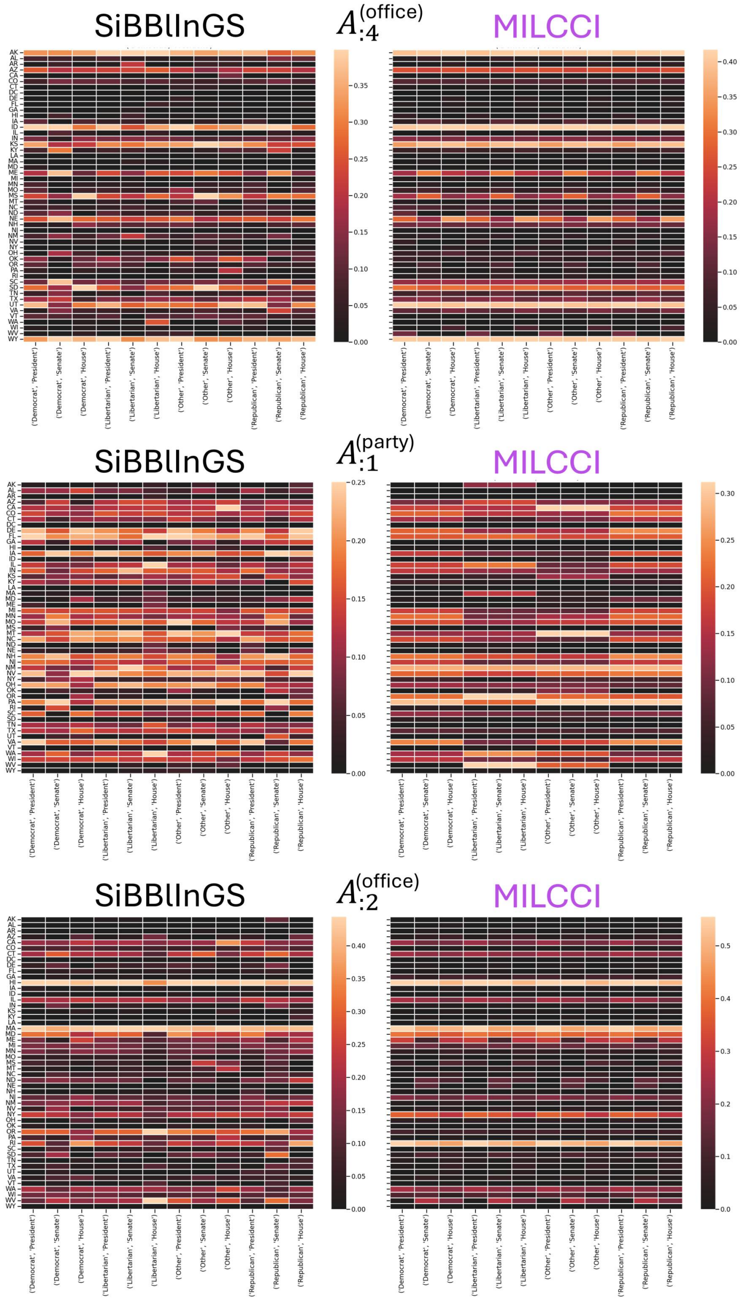


Figure 11: Comparison of components identified by MILCCI and SiBBlInGS for three example labels under the same parameters and seed. MILCCI components were duplicated to match the x-tick labels of SiBBlInGS. SiBBlInGS shows uninterpretable changes across every label, even when parts are shared, whereas MILCCI disentangles them.

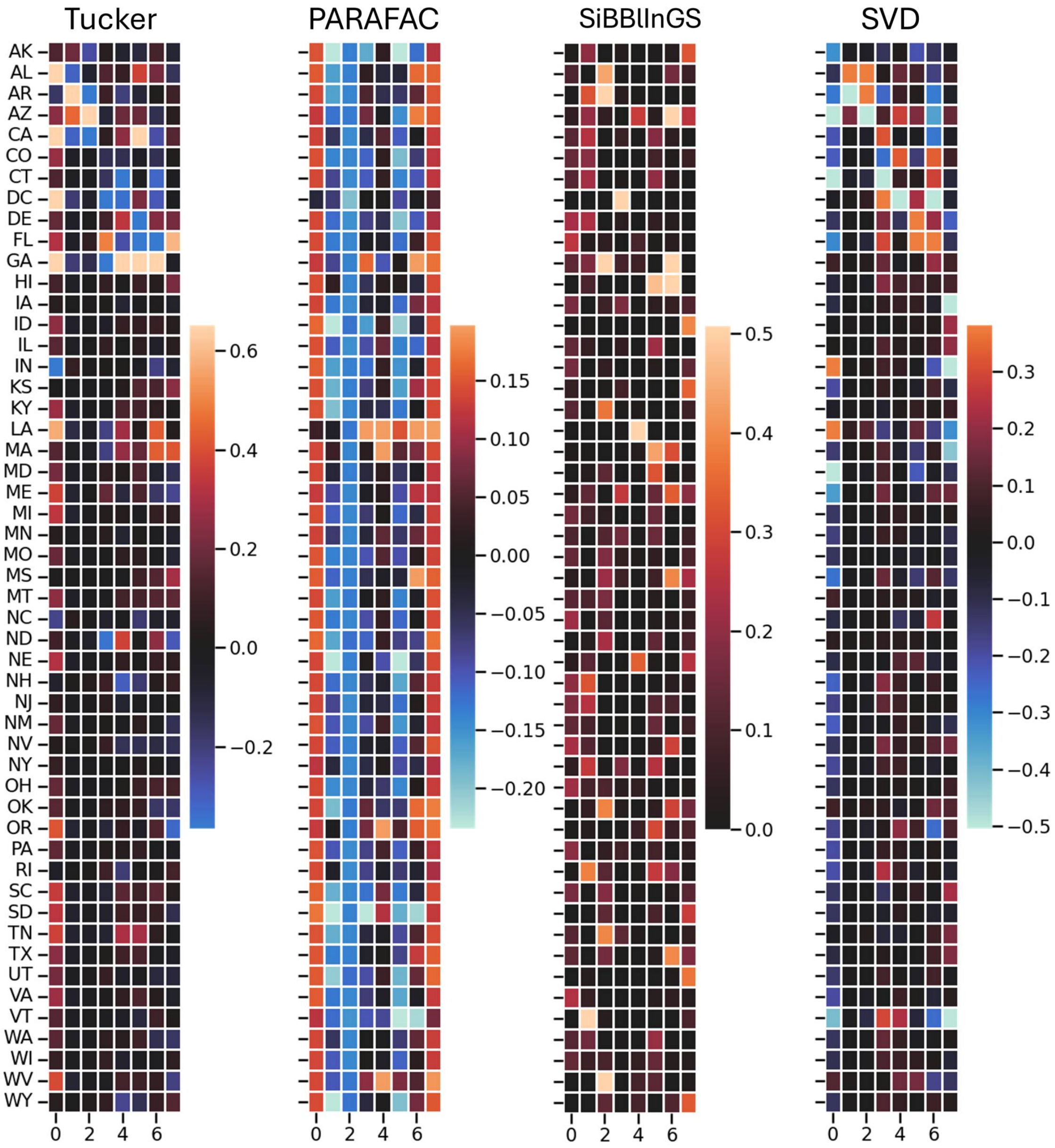


Figure 12: **Voting Experiment Baseline Comparison**. Components identified by the following baselines: 1) Tucker Decomposition (HOSVD), 2) PARAFAC, 3) SiBBlInGS (for a single random label entry: (Democrat, President)), 4) SVD (on all concatenated trials).

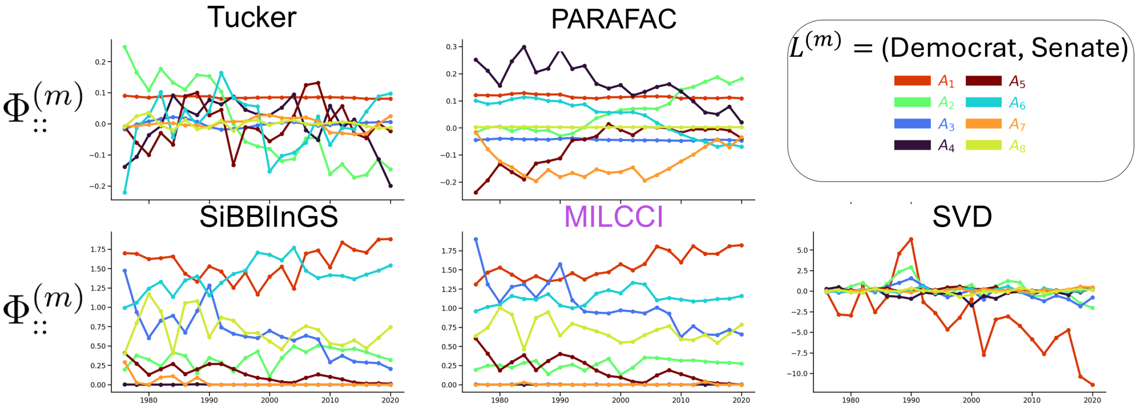


Figure 13: **Voting Experiment**. Traces Identified by MILCCI compared to the other baselines for example random trial.

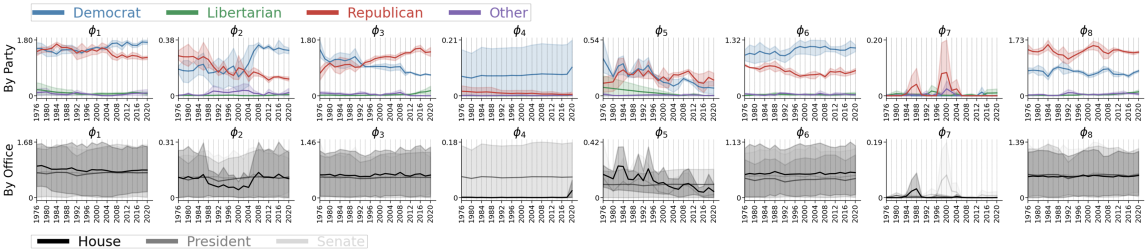


Figure 14: **Voting Traces.** Top: Colored by Party. Bottom: Colored by Office.

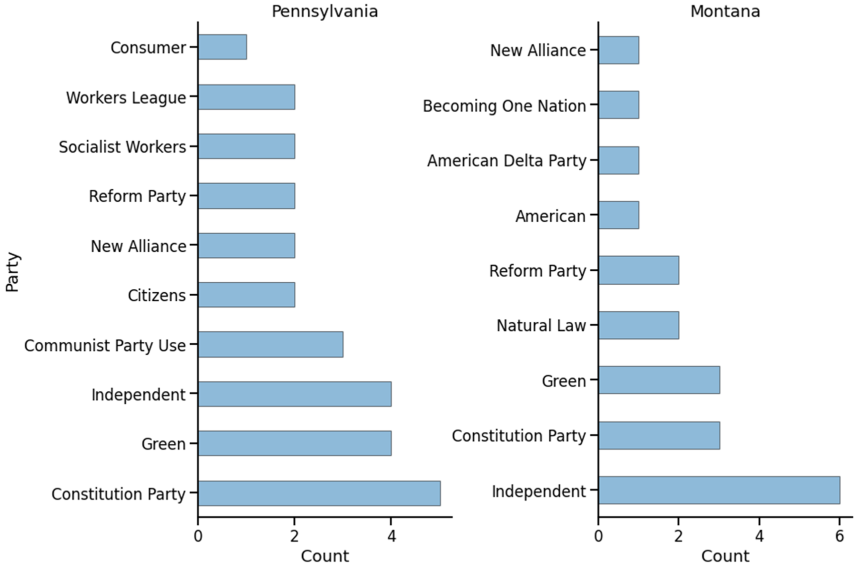


Figure 15: Top “Other” parties instance counts for Montana and Pennsylvania

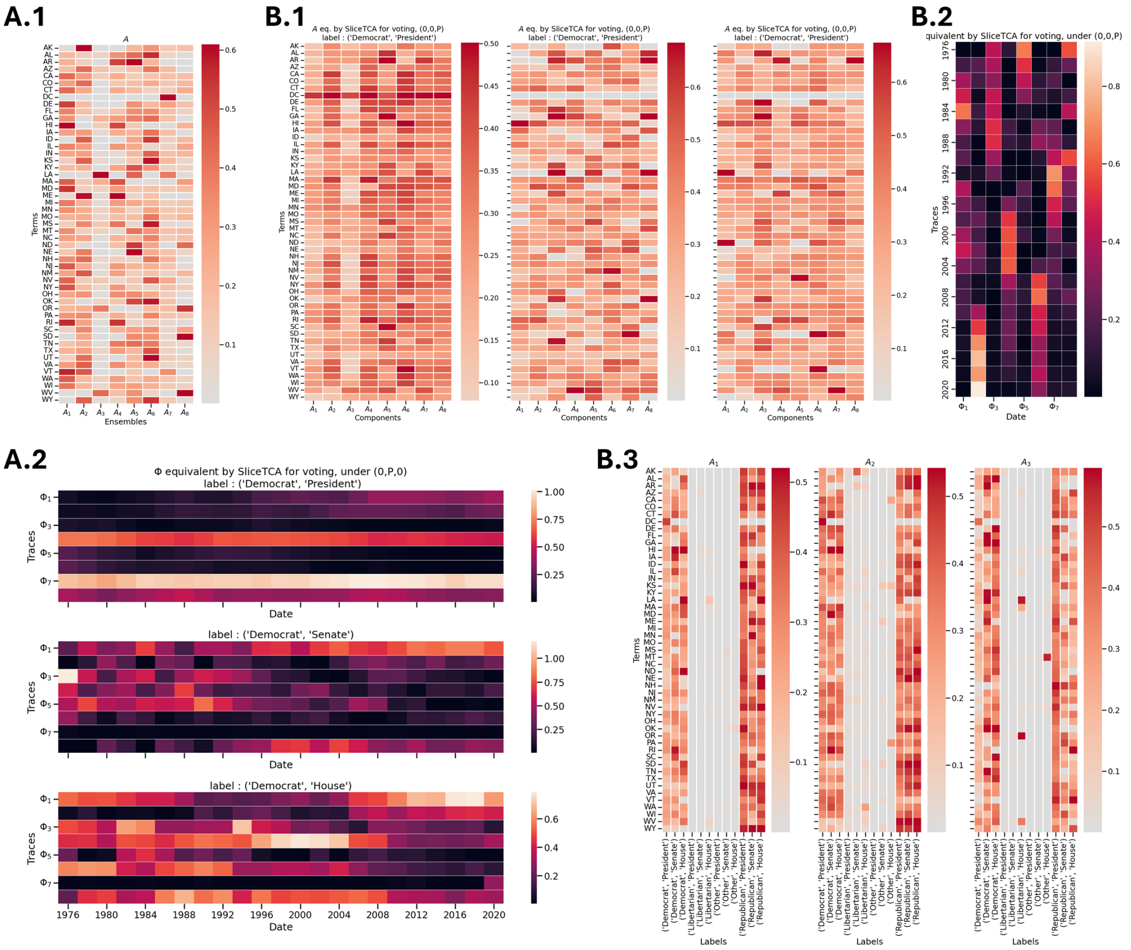


Figure 16: **Voting experiment: SliceTCA comparison.** **A.1** Components from the fixed component case (i.e., from u). **A.2** Traces from the fixed component case (i.e., from A). **B.1** Components from the varying component case (i.e., B). Each subplot represents a trial and columns show its different components. **B.2** Components from the varying component case (i.e., v). **B.3** Components from the varying component case. Each subplot represents one component and columns represent trials. We can observe that there is no consistency within the same component (B.3) across trials compared to across components.

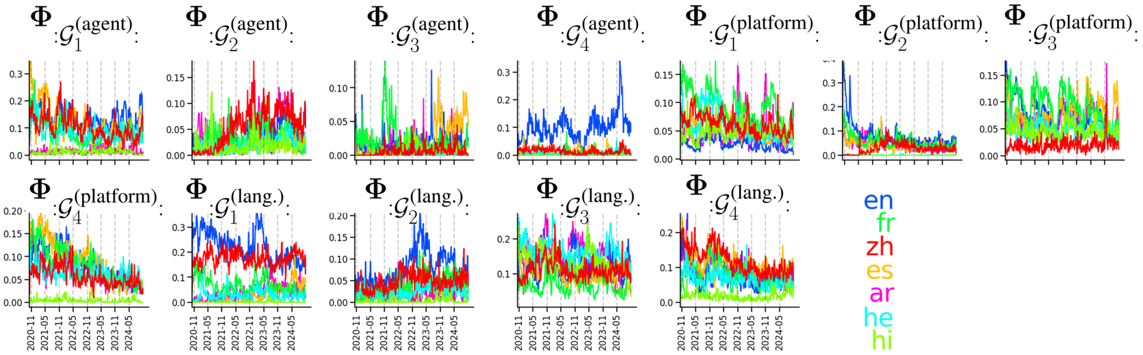


Figure 17: **Wikipedia Page View Experiment**. Traces colored by Language (Arabic, English, Espagnole, French, Hebrew, Hindi, Chinese) Across All Ensembles.

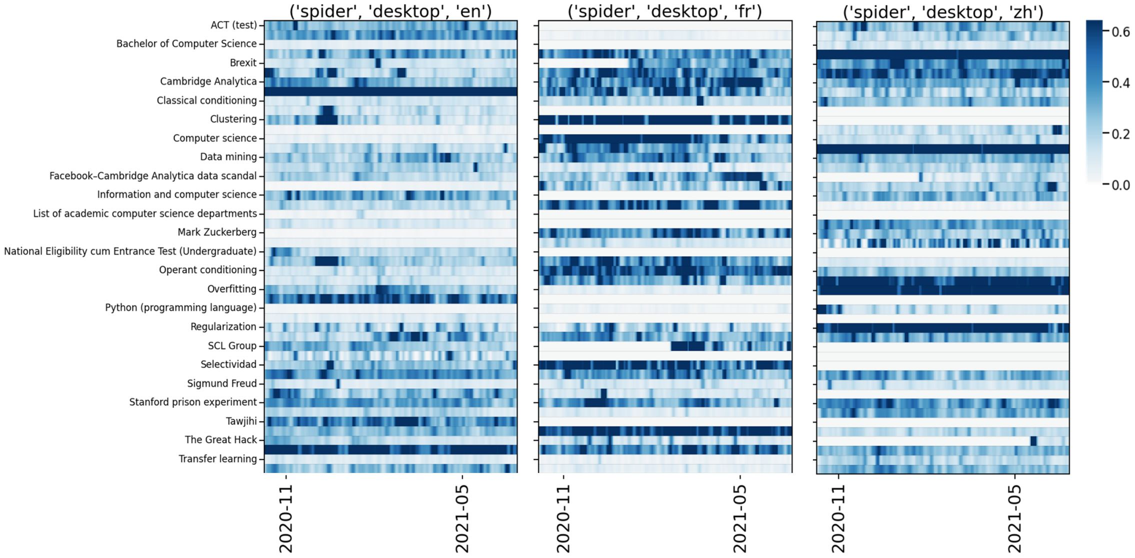


Figure 18: Wikipedia Page View Data, example Trials

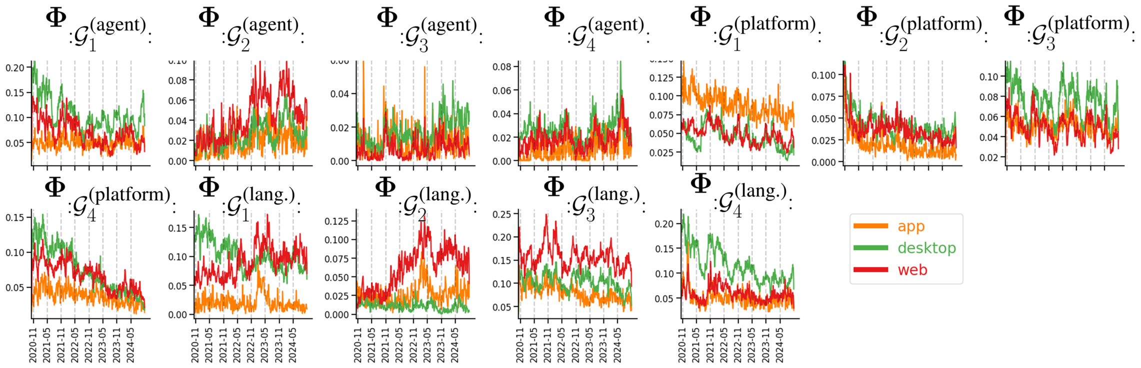


Figure 19: **Wikipedia Page View Experiment**. Traces colored by Platform (desktop, mobile web, mobile app) Across All Ensembles

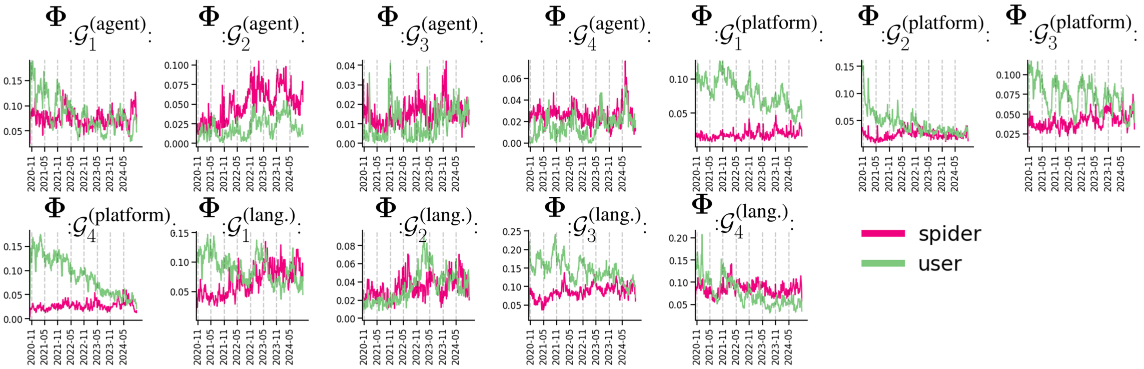


Figure 20: **Wikipedia Page View Experiment**. Traces colored by agent (Spider vs. User) Across All Ensembles

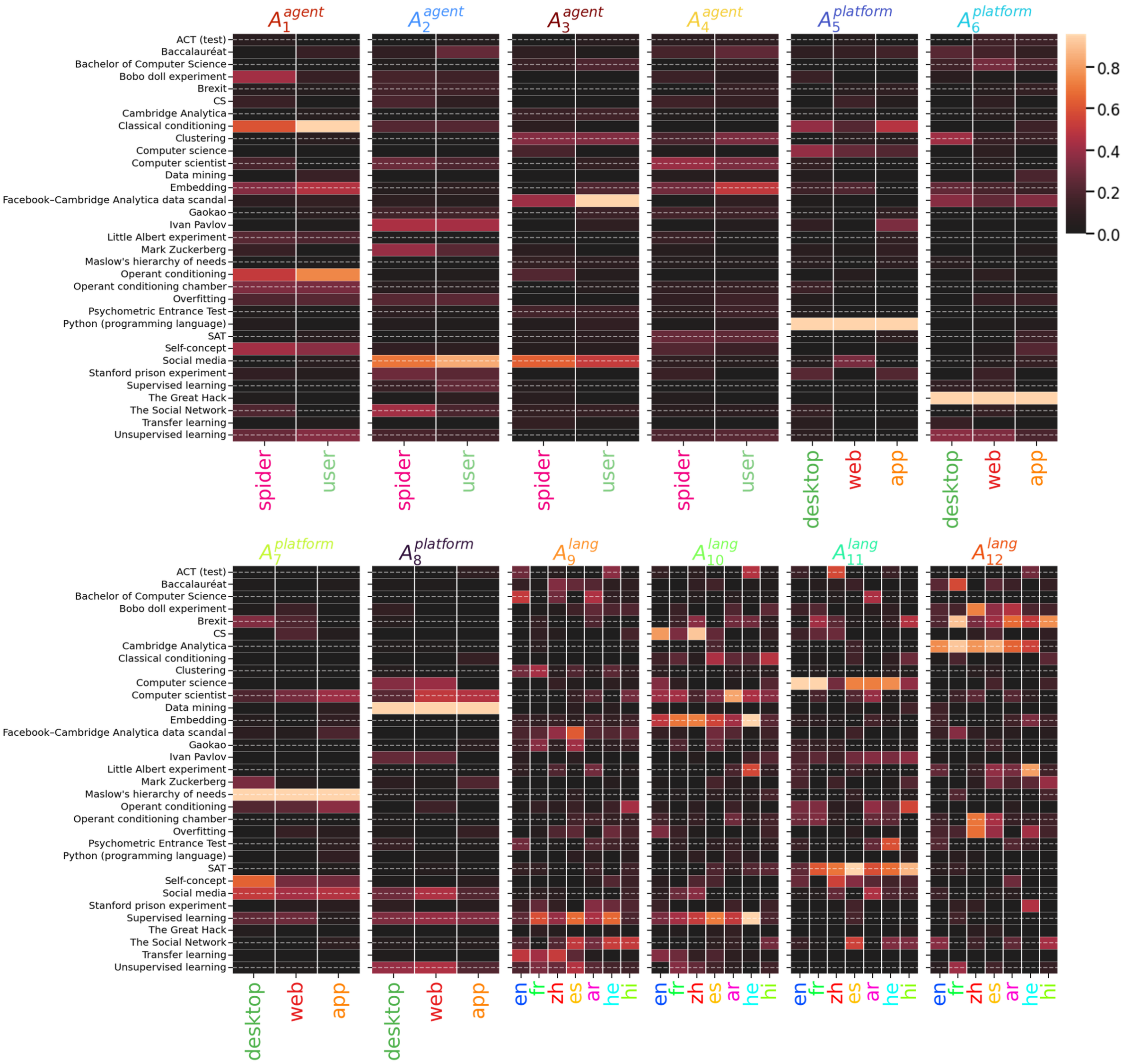


Figure 21: Components identified for Wikipedia Page-view Experiment.

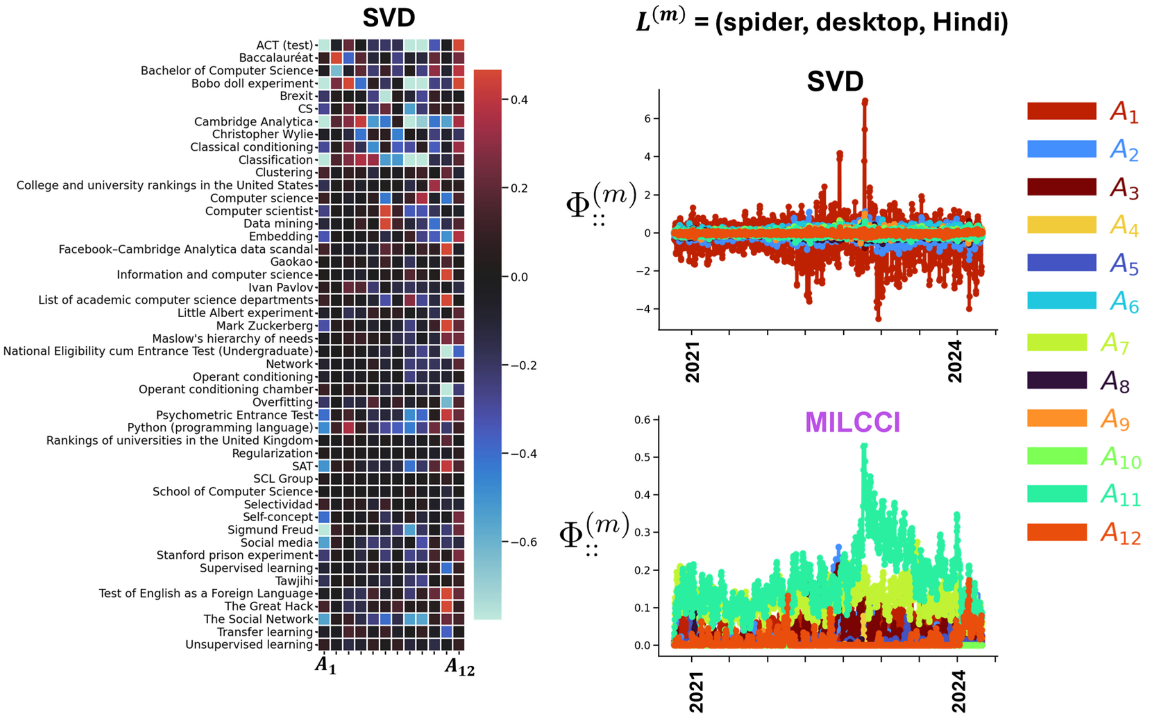


Figure 22: **Wikipedia Experiment Compared to SVD**. Components identified by SVD (compositions on the left, example trial traces on the right).

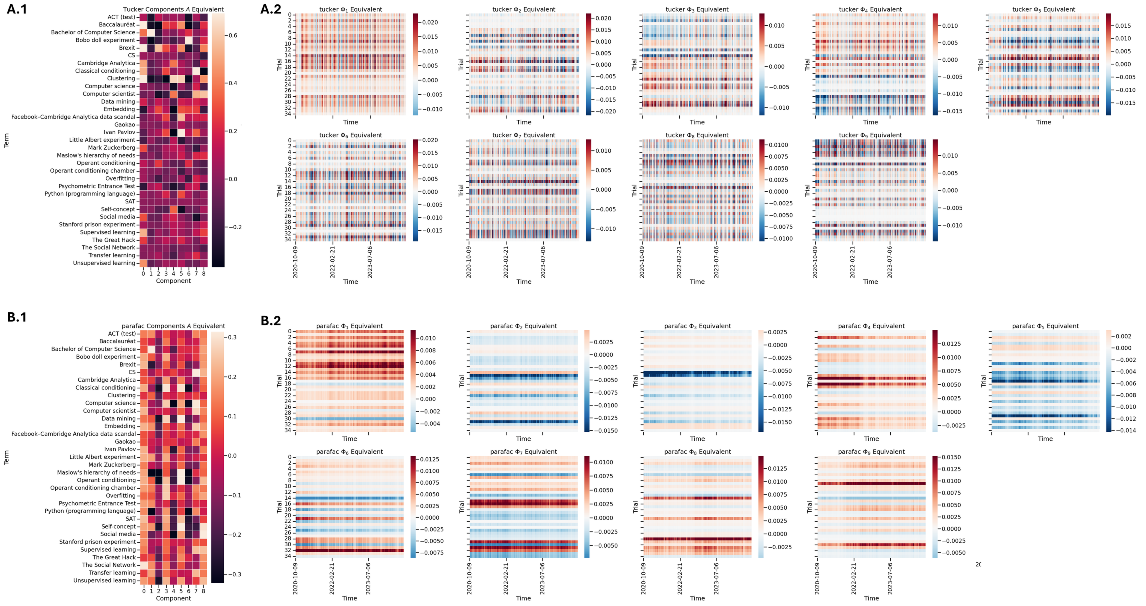
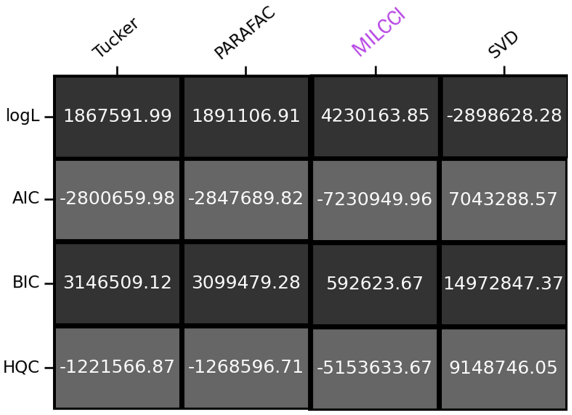


Figure 23: Components Identified by Tucker (**A.1**, **A,2**) and PARAFAC (**2.1**, **B.2**) for the Wikipedia Experiment, see App. H



	Tucker	PARAFAC	MILCCI	SVD
logL	1867591.99	1891106.91	4230163.85	-2898628.28
AIC	-2800659.98	-2847689.82	-7230949.96	7043288.57
BIC	3146509.12	3099479.28	592623.67	14972847.37
HQC	-1221566.87	-1268596.71	-5153633.67	9148746.05

Figure 24: Information Criteria for Wikipedia Experiment, MILCCI vs. baselines. Notably, PARAFAC and Tucker did not converge of 12 components due to SVD instability. Notably, PARAFAC and Tucker encountered instability issues, and hence the results presented for Tucker and PARAFAC here are for rank 9 and added noise with $\sigma = 0.1$ (Tucker) and $\sigma = 0.2$ (PARAFAC).

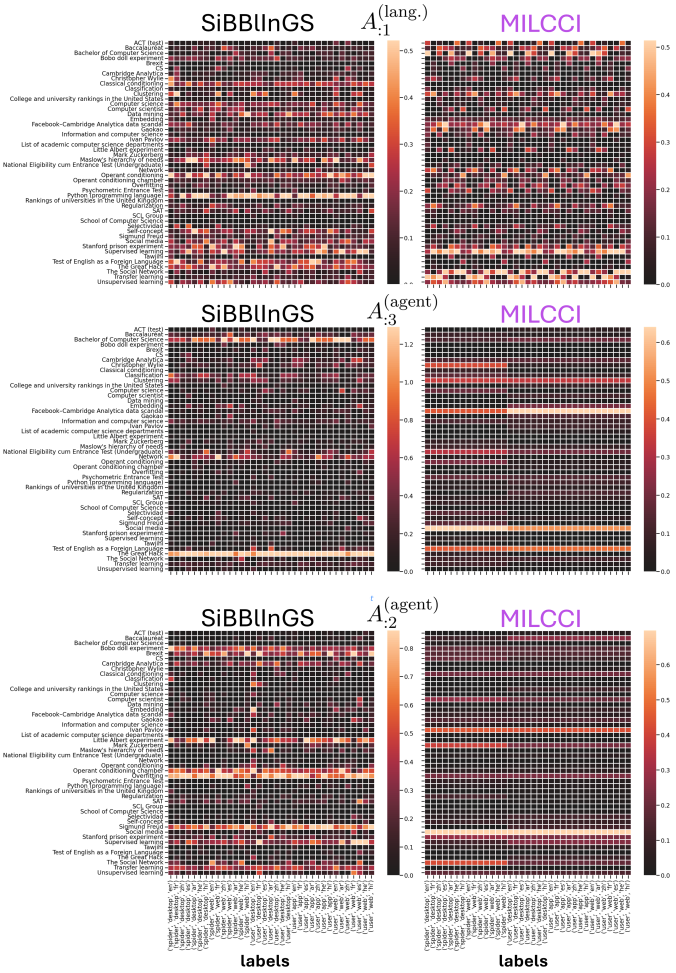


Figure 25: **Wikipedia Components, MILCCI vs. SiBBlInGS**. SiBBlInGS components display compositional changes scattered across labels, rather than the category-specific adjustments captured by MILCCI. *Note:* MILCCI components are shown here with duplicate columns to align with SiBBlInGS components for visualization.

]

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 2: List of US States and Their Abbreviations

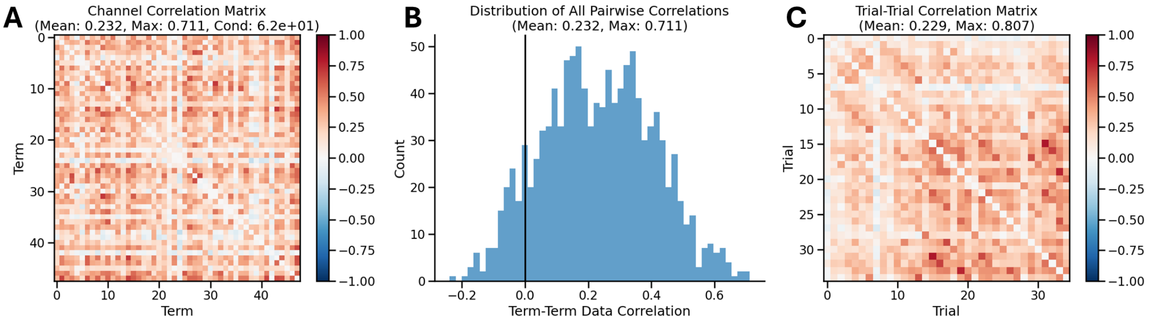


Figure 26: Correlation analysis of Wikipedia dataset reveals moderate correlations that do not explain baseline convergence failures. (A) Channel (term-term) correlation matrix shows moderate correlations with maximum of 0.711 and good condition number (6.2e+01). (B) Distribution of all pairwise correlations demonstrates that most correlations are moderate, with no high correlations (>0.8). (C) Trial-trial correlation matrix shows similar moderate correlation patterns (max 0.807). The data has full rank (48/48) and reasonable correlation structure, indicating that baseline convergence failures (Tucker, PARAFAC) stem from algorithmic limitations rather than problematic data characteristics.

Table 3: 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

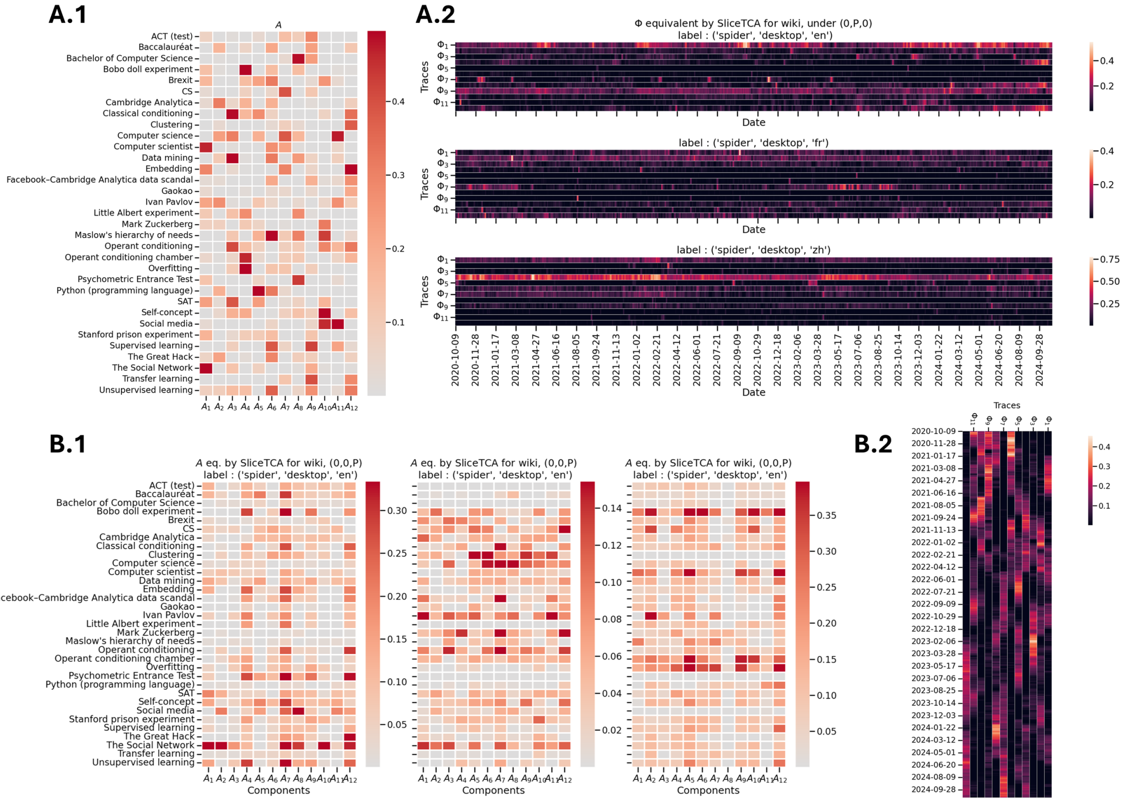


Figure 27: **Wikipedia experiment: SliceTCA comparison.** Components equivalent to MILCCI's were extracted from SliceTCA's configuration 1 (see Sec. H.2), i.e., components extracted from sliceTCA's $\boldsymbol{u}$ vector (**A.1**) and temporal traces from sliceTCA's $\boldsymbol{A}$ matrix (**A.2**). For the second configuration (**B** panels), components (MILCCI's $\boldsymbol{A}$) were extracted from SliceTCA's $\boldsymbol{C}$ (**B.1**) and traces from SliceTCA's $\boldsymbol{v}$ (**B.2**). See details in Section H.2.

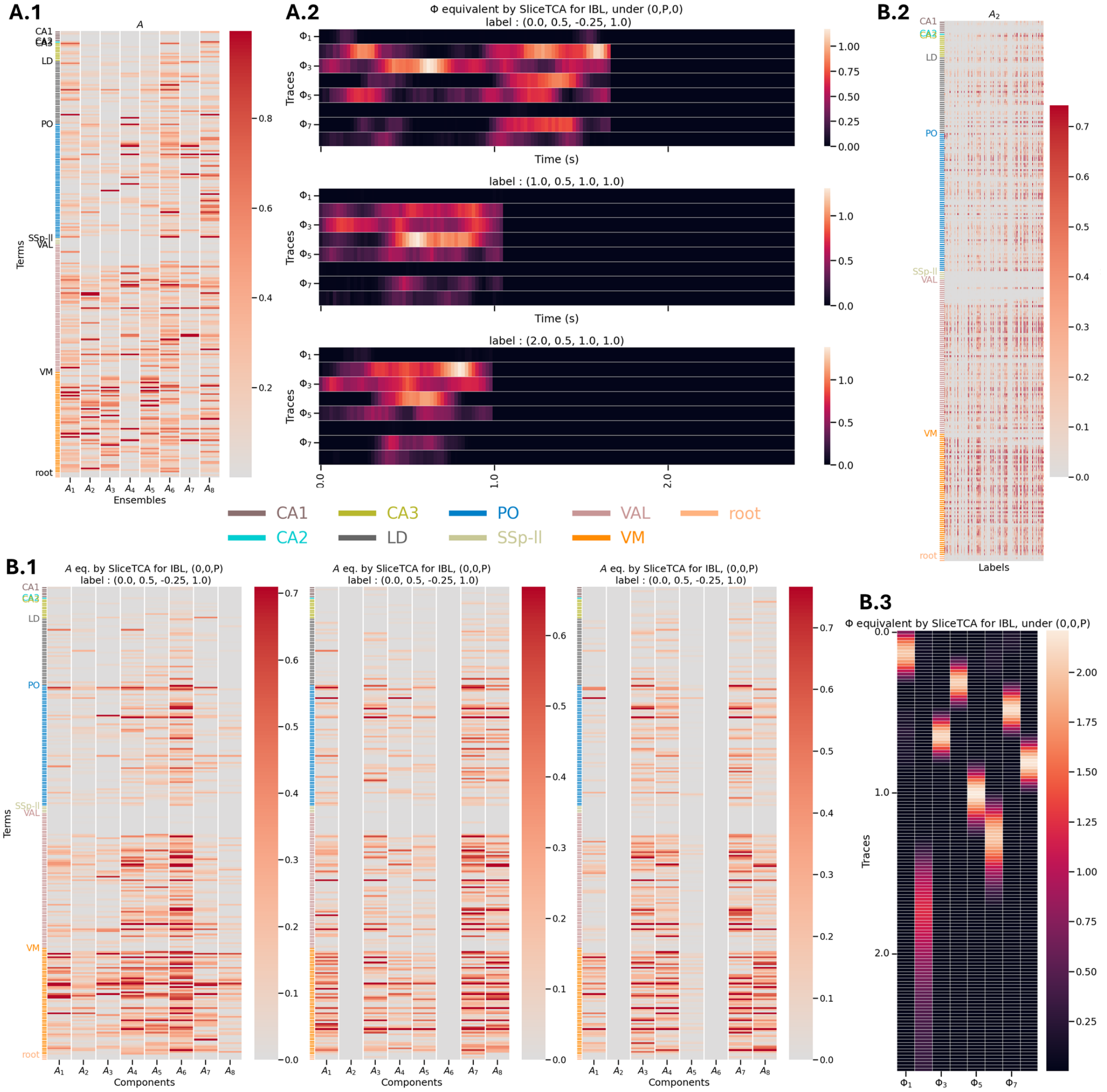


Figure 28: **IBL experiment: SliceTCA components. A:** sliceTCA's configuration 1 (see Sec. H.2). **A.1:** Components from the fixed component case (i.e., from sliceTCA's $\boldsymbol{u}$). **A.2:** Traces from the fixed component case (i.e., from sliceTCA's $\boldsymbol{A}$).

B: sliceTCA's configuration 2: **B.1:** Components from the varying component case. **B.2:** Components showing how they vary over labels (example: component 2). **B.3:** Traces from the sliceTCA's $\boldsymbol{v}$ vector. See details in Section H.2.

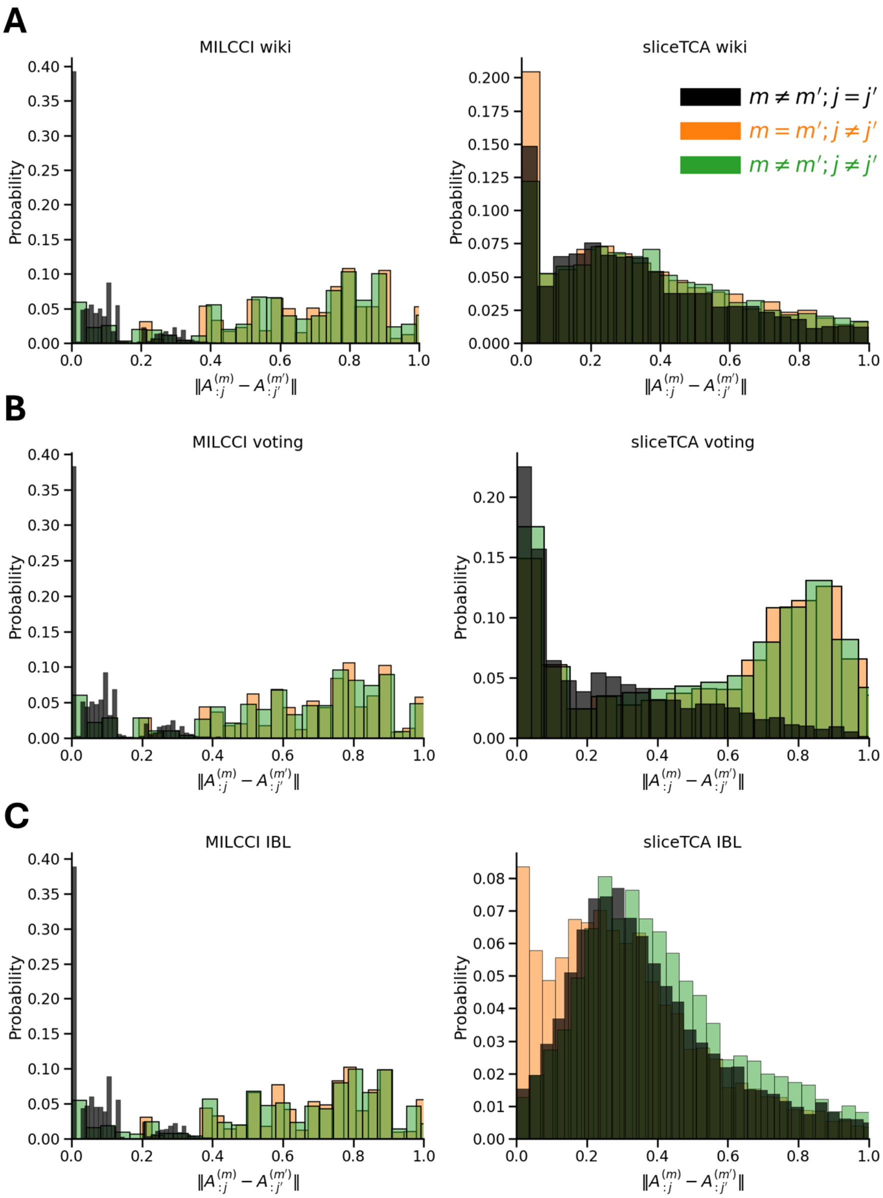


Figure 29: **Distribution of distances between component pairs.** Comparison of distances between same components (black) versus different component pairs across three datasets and two methods, presented for sliceTCA configuration 2 that allow components change (Sec. H.2). **A** Wiki dataset, **B** Voting dataset, **C** IBL dataset. Left column shows MILCCI results, right column shows sliceTCA results. Black bars represent distances between the same component across different conditions ($m \neq m'; j = j'$), orange bars represent distances between different components in the same condition ($m = m'; j \neq j'$), and green bars represent distances between different components in different conditions ($m \neq m'; j \neq j'$).

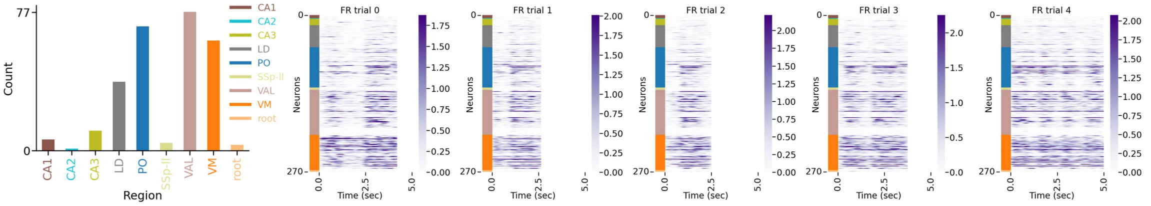


Figure 30: IBL data following our pre-processing steps.

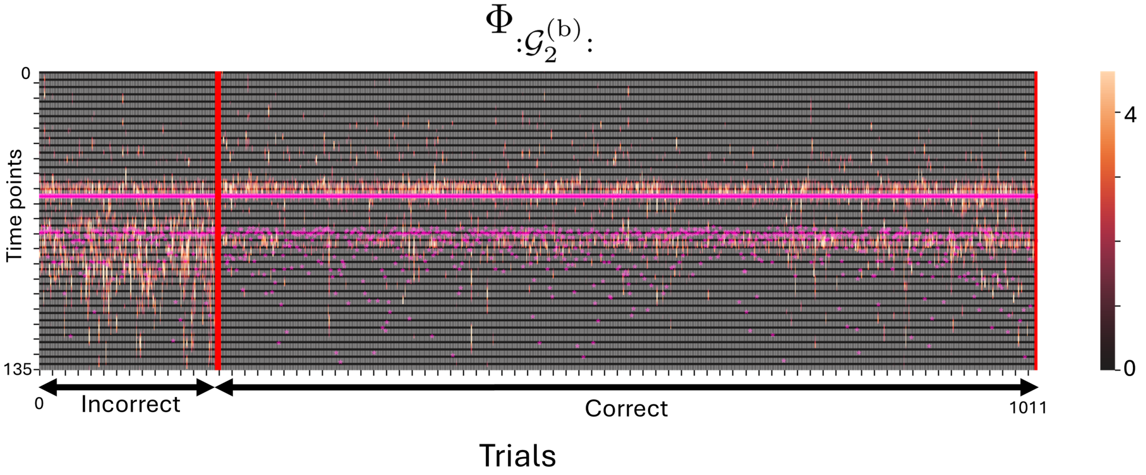


Figure 31: Traces of $\mathbf{\Phi}_{:\mathcal{G}_2^{(\mathrm{b})}:}$ across trials, separated by decision correctness. Solid pink: stimulus on. Dashed pink: stimulus off. The stimulus appears for a median duration of 1.45 s across trials.

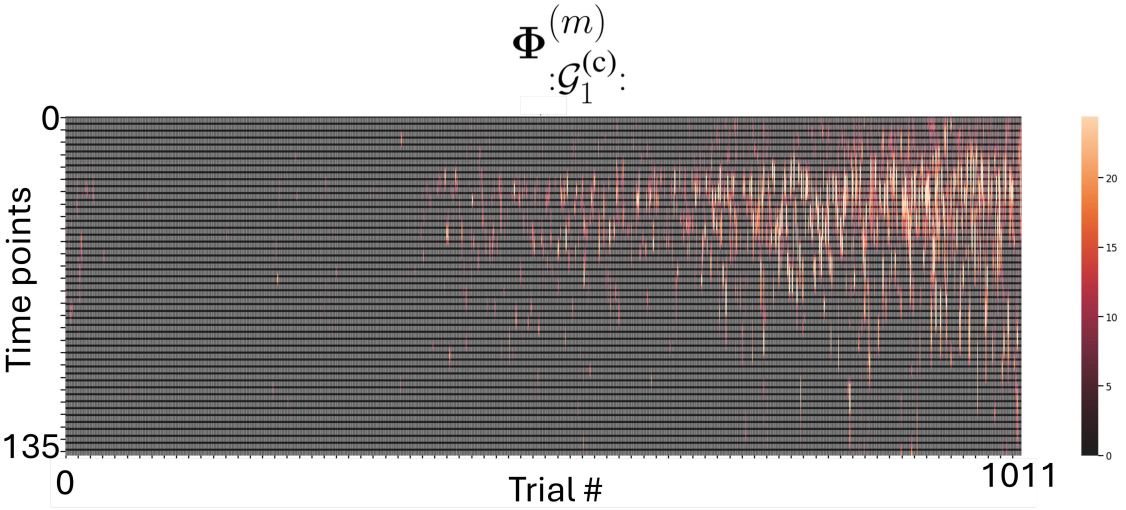


Figure 32: $\mathbf{\Phi}_1^{(\mathrm{c})}$ presents an increasing temporal drift over trials.

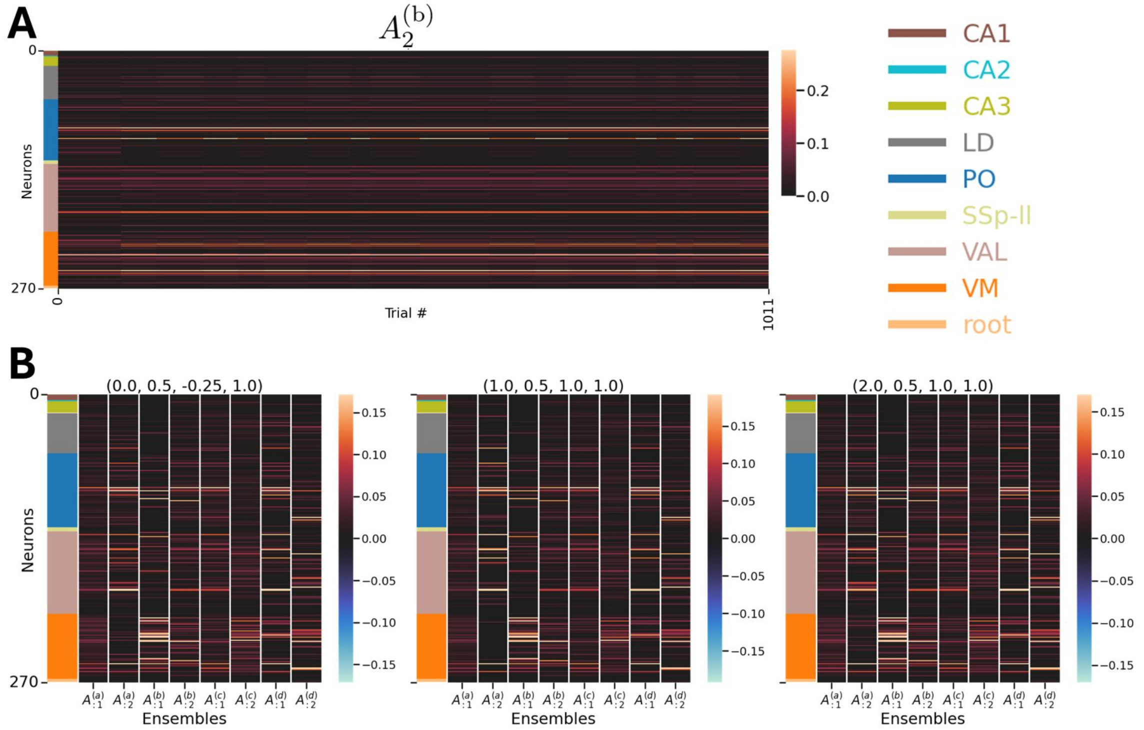


Figure 33: **Components identified by MILCCI in the IBL experiment. A:** Example ensemble (with its trace discussed in the main text) and its adjustments across trials. **B:** Example ensemble matrices reconstructed for three random trials. Each subplot shows all ensembles present in that trial under the unique set of labels indicated.

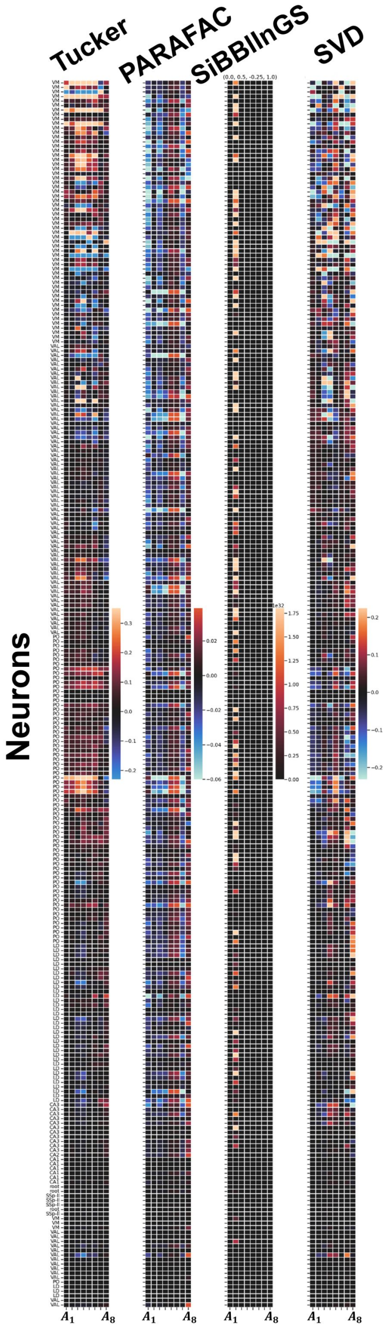


Figure 34: **IBL Neuronal Ensembles Components Identified by Baselines.**

E Additional Information—Voting Experiment

E.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 model to capture two following categories: 1) party (4 options), and 2) office (3 options). We ran MILCCI with $p^{(\mathrm{k})} = 4$ 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.

E.2 Additional Findings—Voting Data

Some more interesting voting patterns, beyond these discussed in the main text, can be found in Figs. 14 and 10.

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. 14 top), Democrat vs. Republican is the dominant axis, consistently driving the most prominent distinctions over time. In some traces ($\boldsymbol{\Phi}_{:\mathcal{G}_1^{(\text{party})}}$, $\boldsymbol{\Phi}_{:\mathcal{G}_2^{(\text{party})}}$, $\boldsymbol{\Phi}_{:\mathcal{G}_3^{(\text{party})}}$), 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 $\boldsymbol{\Phi}_{:\mathcal{G}_2^{(office)}:}$ and $\boldsymbol{\Phi}_{:\mathcal{G}_4^{(office)}}$, show broader fluctuations over time in opposite directions, which may reflect deeper, long-standing historical or structural differences between the parties. Trace $\boldsymbol{\Phi}_{:\mathcal{G}_3^{(office)}}$ 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. 14) 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.3 Post-hoc Validation Analysis on Voting Data

To validate that MILCCI discovers genuine voting structure rather than spurious correlations, we perform comprehensive statistical analyses that examine individual state contributions and test against multiple null hypotheses (Fig. 38).

We remove each state from the component matrix by zeroing out that state's contribution and measure reconstruction degradation. For each of the N states, we calculate reconstruction MSE with the modified components where the target state is excluded. Fig. 38A shows the reconstruction MSE when each state is removed, with the baseline MSE (dashed line) that represents performance with all states included. Fig. 38B displays each state's contribution calculated as the percentage change in MSE relative to baseline performance.

We calculate each state's fair contribution with game-theoretic Shapley values, which consider all possible combinations of states. Due to computational constraints with N=52 states (which require 2^{52} calculations), we use approximation with 500 random combinations. For each combination, we include only combination members in the components matrix while we zero non-combination states, then calculate each state's marginal contribution as the difference in reconstruction quality when that state is added to the combination. Fig. 38C shows the Shapley values, which quantify each state's individual contribution to overall reconstruction quality.

We test three null hypotheses by comparison of original reconstruction performance against randomized versions:

Shuffle Rows (Fig. 38D, left):
We randomly reassign complete state voting profiles to different state positions. This tests whether the specific correspondence between geographic states and their voting behavioral patterns is meaningful.

Random Control (Fig. 38D, middle):
We replace the entire components matrix with random values drawn from a normal distribution with the same global mean and standard deviation as the original data. This tests against pure statistical noise baseline.

Shuffle Each Component (Fig. 38D, right):
For each component dimension, we randomly permute the assignment of states within that component while we preserve component-wise statistics. This tests whether the specific coordination between components within each state matters.

Each permutation test runs 1000 iterations, with p-values calculated as the fraction of permuted reconstructions that achieve equal or better performance than the original.

Component-specific permutation results (Fig. 38E) demonstrate that discovered patterns are robust across individual voting components. We store intermediate reconstruction results for each component, which allows examination of component-specific robustness to randomization. Fig. 38F provides the state abbreviation key for reference.

All statistical tests yield p $<$ 0.001, which provides evidence that MILCCI's discovered voting patterns represent genuine structure.

F Additional Information–Wikipedia Experiment

F.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).

F.2 Clarification on Findings—Wikipedia Data

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

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

G 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).

H Information About Baseline Calculation and Execution

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

H.1 Comparison to SVD, Tucker, PARAFAC

We compared these methods to MILCCI both quantitatively and qualitatively. For the qualitative comparison, we provide example 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. These errors often occur due to (1) many missing values (though no NaNs exist in our data), or (2) high-resolution (daily) measurements introducing highly correlated (Fig. 26) or nearly linearly dependent structures in the data. Hence, for the Wikipedia comparison, in addition to comparisons to SVD, SiBBlInGS, and sliceTCA, we tested Tucker and PARAFAC under lower ranks with increasing added noise. We found that these models converge at rank 9 with added i.i.d. Gaussian noise ($\sigma = 0.1$ for Tucker and $\sigma = 0.2$ for PARAFAC), and the results presented here are under these conditions.

H.2 Comparison to SliceTCA

We note that a direct comparison between our method and SliceTCA (Pellegrino et al., 2024) is limited since SliceTCA is not tailored to find subtle label-driven supervised reorganization patterns in neural ensembles. SliceTCA performs an unsupervised decomposition $\hat{\boldsymbol{X}}_{n,t,k} = \sum_{r=1}^{R_{\text{neuron}}} u_n^{(r)} A_{t,k}^{(r)} + \sum_{r=1}^{R_{\text{time}}} v_t^{(r)} B_{n,k}^{(r)} + \sum_{r=1}^{R_{\text{trial}}} w_k^{(r)} C_{n,t}^{(r)}$ that finds R components. In their notation, $\boldsymbol{u}^{(r)}$ (neural loading vector) is equivalent to one column of our $\boldsymbol{A}$ (i.e., $\boldsymbol{A}_{:,j}$), while their slice $\boldsymbol{A}^{(r)}$ (time-by-trial matrix) would correspond to our $\boldsymbol{\Phi}$ for all trials. Similarly, $\boldsymbol{v}^{(r)}$ (time loading vector) combined with $\boldsymbol{B}^{(r)}$ (neuron-by-trial slice) provides an alternative decomposition. SliceTCA's R_{neuron} corresponds to our P.

We compared MILCCI to SliceTCA (Pellegrino et al., 2024) with $R_{\text{neuron}} = P$ using their publicly available implementation at `https://github.com/arthur-pe/slicetca`. We followed the parameters outlined in their Google Colab notebook: positive=True, learning_rate=5×10^{-3}, min_std=10^{-5}, max_iter=1,000 for Wiki and Synth, 5,000 for IBL, and seed=0.

We tested two separate configurations as baselines:

1. **Configuration (1):**
   considers one neural component (e.g., one ensemble) that varies its traces over trials without additions, which is the closest to MILCCI in terms of formulation, but not enabling small changes in ensembles.
2. **Configuration (2):**
   uses a matrix of neurons × trials (i.e., captures how neurons can change over trials) via the matrix $\boldsymbol{B}$, however each matrix of neurons by trial has one trace. This captures mainly the ensemble adjustment to trial in an unconstrained way (unlike MILCCI) and enables more flexibility in that, but on the other hand restricts the traces more. Any other combination of ranks would not be interpretable in terms of comparison to MILCCI.

We extracted MILCCI-equivalent $\{\boldsymbol{A}\}$ and $\boldsymbol{\Phi}$ as follows:

1. **Configuration (1)**:
   The $\boldsymbol{u}^{(r)}$ vectors form the ensemble matrix (fixed across trials), while temporal traces are extracted from the $\boldsymbol{A}^{(r)}$ matrices by breaking to trials.
2. **Configuration (2)**:
   The ensembles are captured by the rows of $\boldsymbol{B}^{(r)}$ and their variation over trials by the columns. Traces are given by $\boldsymbol{v}^{(r)}$.

H.3 SiBBlInGS

:
MILCCI's main advantage over SiBBlInGS (Mudrik et al., 2024) 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 increase 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). In contrast, MILCCI can handle all of these without requiring additional dimensions, and can also account for the ordinal nature of each category, whereas SiBBlInGS requires choosing a single sorting across all categories. While we acknowledge quantitative metrics would also be valuable against SiBBlInGS, SiBBlInGS' graph-driven sparsity makes standard model comparison metrics (AIC, BIC) irrelevant. Particularly, SiBBlInGS inference includes a graph-based reweighting sparsity mechanism that hinders accurate estimation of degrees-of-freedom needed for information criteria calculations, making information criteria comparison intractable. Hence, we limited quantitative comparisons against SiBBlInGS

to synthetic data (Fig. 2) and, for the real-world experiments, we focused on qualitative comparisons that emphasize MILCCI's interpretability advantages.

I Alternative Inference of traces via Dynamic Prior

In the main text, we regularize the temporal traces $\boldsymbol{\Phi}^{(m)}$ using a smoothness penalty. Here, we present an alternative formulation where the temporal traces evolve according to a Linear Dynamical System (LDS), suitable for data with non-stationary dynamics.

I.1 Linear Dynamical System Prior

We assume that for each trial m, the temporal traces follow:

$$\boldsymbol{\phi}_t^{(m)} = \boldsymbol{W}^{(m)} \boldsymbol{\phi}_{t-1}^{(m)} + \boldsymbol{\eta}_t, \quad t = 2, \ldots, T^{(m)} \tag{4}$$

where $\boldsymbol{W}^{(m)} \in \mathbb{R}^{P \times P}$ is a trial-specific transition matrix and $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$.

I.2 Modified Objective and Inference

We modify the optimization to jointly learn traces and dynamics. The algorithm alternates between:

Step 1: Update $\boldsymbol{A}^{(k)}$ (for each category k)

Same as main text, using Equation 2.

Step 2: Update $\{\boldsymbol{\Phi}^{(m)}, \boldsymbol{W}^{(m)}\}$ (for each trial m)

Given fixed $\tilde{\boldsymbol{A}}$, we perform inner iterations (3-5 times):

Step 2a: Update $\boldsymbol{\Phi}^{(m)}$ given $\boldsymbol{W}^{(m)}$

$$\hat{\boldsymbol{\Phi}}^{(m)} = \arg\min_{\boldsymbol{\Phi}^{(m)}} \left\| \boldsymbol{Y}^{(m)} - \tilde{\boldsymbol{A}}(\boldsymbol{L}^{(m)}) \boldsymbol{\Phi}^{(m)} \right\|_F^2 + \gamma_3 \sum_{t=2}^{T^{(m)}} \left\| \boldsymbol{\phi}_t^{(m)} - \boldsymbol{W}^{(m)} \boldsymbol{\phi}_{t-1}^{(m)} \right\|_2^2 + \gamma_4 \left\| (\boldsymbol{C} \odot (\mathbf{1} - \boldsymbol{I}_P)) \odot \boldsymbol{D} \right\|_{1,1} \tag{5}$$

Step 2b: Update $\boldsymbol{W}^{(m)}$ given $\boldsymbol{\Phi}^{(m)}$

With regularization $R(\boldsymbol{W}^{(m)}) = \|\boldsymbol{W}^{(m)} - \boldsymbol{I}\|_F^2$ to encourage stability:

$$\hat{\boldsymbol{W}}^{(m)} = \left(\sum_{t=2}^{T^{(m)}} \boldsymbol{\phi}_t^{(m)} (\boldsymbol{\phi}_{t-1}^{(m)})^T + \gamma_5 \boldsymbol{I} \right) \left(\sum_{t=2}^{T^{(m)}} \boldsymbol{\phi}_{t-1}^{(m)} (\boldsymbol{\phi}_{t-1}^{(m)})^T + \gamma_5 \boldsymbol{I} \right)^{-1} \tag{6}$$

The inner iterations stabilize both $\boldsymbol{\Phi}^{(m)}$ and $\boldsymbol{W}^{(m)}$ before updating $\boldsymbol{A}$, preventing noise amplification.

I.3 Initialization

1. Initialize $\boldsymbol{\Phi}^{(m)}$ using the original smoothness-based objective (Equation 3)

2. Initialize $\boldsymbol{W}^{(m)} = \left(\sum_t \boldsymbol{\phi}^{(m)}_{t,\text{init}}(\boldsymbol{\phi}^{(m)}_{t-1,\text{init}})^T\right)\left(\sum_t \boldsymbol{\phi}^{(m)}_{t-1,\text{init}}(\boldsymbol{\phi}^{(m)}_{t-1,\text{init}})^T\right)^{-1}$

This dynamic prior is suitable for cases that are assumed to be stationary and governed by a single LDS that does not change over time. Notably, real-world data is often non-stationary, and hence this dynamic prior would benefit from extensions in future work, such as learning dynamics like those exemplified in Chen et al. (2024); Linderman et al. (2016); Mudrik et al. (2025)).

Table 4: Timing results (in seconds) for different methods across three experiments. Dash (-): method did not converge for the same dimension.

Experiment	SVD	Tucker	parafac	parafac_scaled	SliceTCA	SiBBlInGS	MILCCI
Voting	20.01	20.12	20.26	20.35	27.58	27.19	28.16
Wiki	20.37	-	-	-	34.05	29.48	31.12
IBL	24.03	268.94	1056.20	1256.74	1000	1004.32	1000.32

J MILCCI identifies neural ensembles that adjust to arousal level and stimulation frequency and evolve via dynamical rules

J.1 Data and Pre-processing

We used one experimental session from Papadopoulos et al. (2024) in NWB format from the DANDI archive Rübel et al. (2022) (DANDI:000986), which the McCormick laboratory at the University of Oregon collected. The dataset contains recordings from mouse auditory cortex during passive exposure to auditory stimuli. The subject was a male mouse (Mus musculus), aged P79D from birth (see experimental illustration taken from Papadopoulos et al. (2024) in Fig. 35A).

We processed spike trains from the NWB file (raster plot in Figure 35E) to estimate firing rates (FR) using Gaussian convolution with the following parameters: 5 ms time bins, 50 ms Gaussian window, and 5 ms σ (Fig. 35F). For this demonstration, we used the first 500 trials. The processed dataset thus contain 235 neurons across 15 unique experimental conditions that varied throughout the first 500 trials.

J.2 Demonstration of Extended MILCCI With Non-Linear Transformation and Dynamics Prior Over Traces Evolution

This experiment provides a glimpse of MILCCI's extensibility by showcasing two key advances: (1) dynamical evolution of neural ensembles, and (2) nonlinear transformation of FR via tanh (normalizing to range [-1, 1]). This serves as a proof of concept for modeling dynamics alongside MILCCI, paving the way for future extensions with more complex non-stationary dynamics (see App. I for details on dynamics inference).

We defined the following categories: (a) arousal level (based on binarized pupil diameter, Fig. 35C); and (b) stimulation frequency (distribution among the used trials in Fig. 35B).

We ran extended MILCCI with $p_j = 3$ ensembles per category.

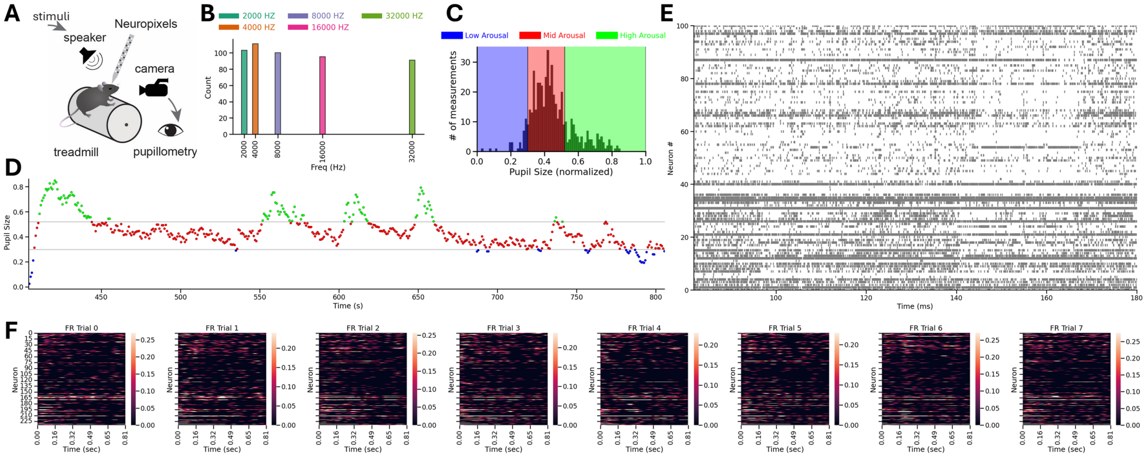


Figure 35: **Additional demonstration of MILCCI on data from (Papadopoulos et al., 2024). A:** Experiment illustration, adapted from (Papadopoulos et al., 2024). **B:** Trial counts per stimulation frequency among the first 500 trials analyzed. **C:** Distribution of normalized pupil size (a.u.) across the first 500 trials. Colored background represents the three arousal states considered. (cutoffs 0.3, 0.52) **D:** Average pupil size per trial. Each marker represents one trial; x-axis shows trial midpoint time from recording start. **E:** Raster plot of the first 180 ms of neural activity. **F:** Example firing rate estimation for the first 8 trials from spike train data.

We leverage this example to demonstrate MILCCI's capacity for extensions, including: 1) pre-processing via non-linearity, and 2) modifying the inference of Φ to evolve via dynamical priors, rather than via Eq. 3 (see Sec. I for details).

J.3 HYPERPARAMETER SENSITIVITY ANALYSIS

To empirically demonstrate MILCCI's robustness to hyperparameter choices, we conducted a comprehensive sensitivity analysis across a wide range of values. We tested 20 values for $\gamma_1 \in [0.002, 0.5]$ and 40 values for $\gamma_2 \in [0.002, 0.5]$, yielding 800 total hyperparameter combinations across multiple ensemble instances.

To quantify differences between learned component matrices $\boldsymbol{A}$, we employed the normalized Frobenius distance:

$$d_{\text{norm}}(\boldsymbol{A}_1, \boldsymbol{A}_2) = \frac{\|\boldsymbol{A}_1 - \boldsymbol{A}_2\|_F}{\sqrt{\|\boldsymbol{A}_1\|_F^2 + \|\boldsymbol{A}_2\|_F^2}} \tag{7}$$

We computed two types of distances to assess hyperparameter sensitivity relative to ensemble variability (Fig. 37B,C,F):

- **Within-ensemble distances**: For each ensemble, we computed pairwise distances between component matrices obtained with different hyperparameter settings. These distances quantify how much the learned components vary due to hyperparameter choices while holding the data fixed.
- **Cross-ensemble distances**: For each hyperparameter setting, we computed pairwise distances between component matrices from different ensemble instances. These distances quantify how much the learned components vary due to data sampling and stochastic initialization while holding hyperparameters fixed.

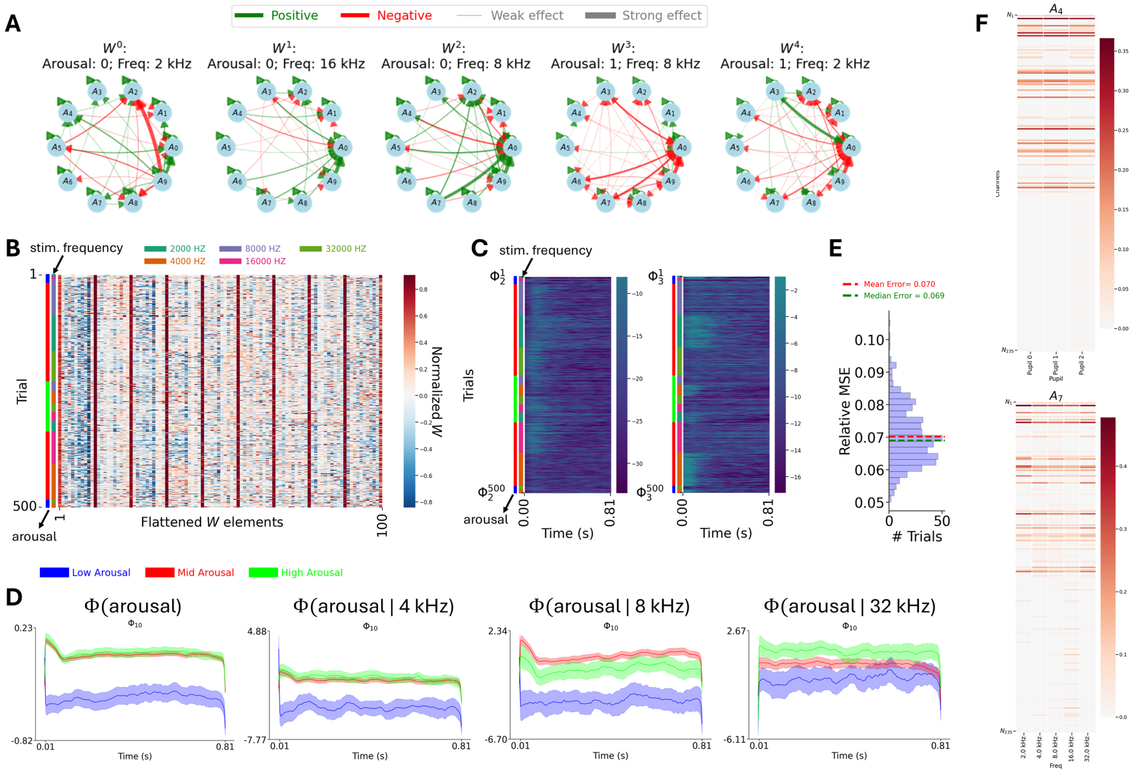


Figure 36: **Demonstrating MILCCI with nonlinear tanh data transformation** ($\tanh\left(\boldsymbol{y}^{(m)}\right) \approx \sum_{(\mathbf{k})\in C} \boldsymbol{A}^{(\mathbf{k})}_{::L^{(m)}_{\mathbf{k}}} \boldsymbol{\Phi}^{(m)}_{\mathcal{G}^{(\mathbf{k})}:}$) **and dynamical constraints over the trace evolution** ($\boldsymbol{\Phi}^{(m)}_t \approx \boldsymbol{W}^{(m)} \boldsymbol{\Phi}^{(m)}_{t-1}$) **, via recent neural data from (Papadopoulos et al., 2024).** **(A)** Transition networks $\boldsymbol{W}^{(m)}$ for five example trials from different arousal and stimulus frequency conditions, showing learned interactions between ensembles. **(B)** Heatmap of all learned transition matrices (each $\boldsymbol{W}^{(m)}$ flattened) across trials (rows). Scatters on the left indicate trial conditions: arousal level (left side) and stimulation frequency (right side). Weights are normalized by column maximum. **(C)** Temporal traces for two example ensembles ($\boldsymbol{\Phi}^1$ and $\boldsymbol{\Phi}^3$) across trials and time, organized by arousal level, demonstrating condition-dependent temporal dynamics. Scatters on the left are the same as in panel **B**. **(D)** Average temporal traces with stderr grouped by arousal level: without frequency conditioning (left), or conditioned on specific stimulus frequencies (three rightmost panels: 4 kHz, 8 kHz, and 32 kHz). **(E)** Distribution of reconstruction error (relative MSE) across trials. Dashed lines indicate mean and median. **(F)** Example neuronal ensembles and how they subtly adjust to labels, showing condition-specific loadings across channels. Top panels show components varying with arousal level; bottom panels show components adjusting to stimulus frequency.

If MILCCI is robust to hyperparameter choices, within-ensemble distances should be substantially smaller than cross-ensemble distances. This would indicate that the learned components are more sensitive to the underlying data structure than to hyperparameter tuning, demonstrating that the model reliably captures meaningful patterns across reasonable hyperparameter ranges.

Component matrices (Fig. 37A) show that higher γ_1 values produce sparser components as expected, while the overall structure remains stable. Within-ensemble distances are consistently lower than cross-ensemble distances, with a mean ratio of 0.17 (Fig. 37C), indicating that hyperparameter variation introduces less variability than ensemble stochasticity. The distribution of distances (Fig. 37D) reveals separation between the two types of variation. Per-ensemble analyses (Fig. 37B,E) show this pattern holds across all individual ensemble instances, with ratios below 1. These results demonstrate model robustness, as hyperparameter variation contributes minimally compared to stochastic ensemble variation.

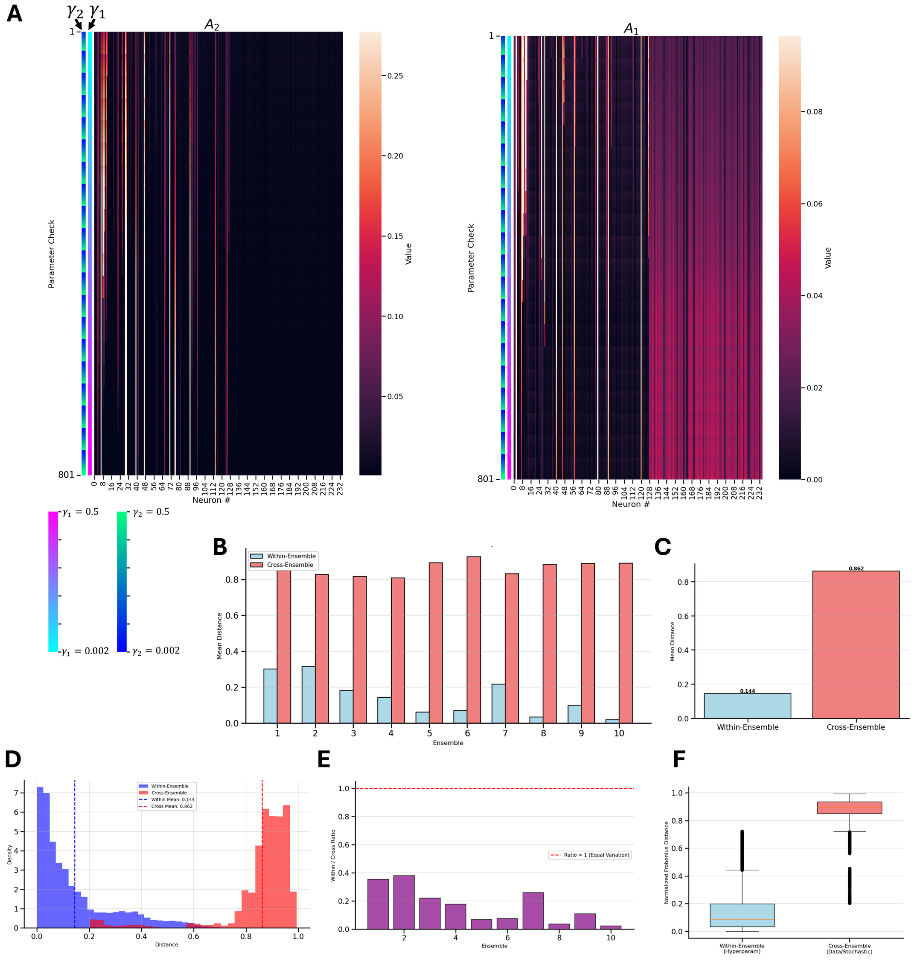


Figure 37: **Hyperparameter sensitivity analysis across 800 parameter combinations (App. J.3). A:** Example component matrices from two ensembles across all hyperparameter settings, ordered by γ_1 and γ_2 (colorbars at the bottom). For example, higher γ_1 values yield sparser components. **B:** Per-ensemble comparison of within-ensemble (blue) versus cross-ensemble (red) distances. **C:** Overall mean distances showing within-ensemble variation is smaller than cross-ensemble variation. **D:** Distribution of normalized Frobenius distances reveals clear separation between within-ensemble and cross-ensemble variation. **E:** Robustness ratios for each ensemble, all below 1 (dashed line indicates equal variation). **F:** Overall boxplot comparison confirms systematic difference between within-ensemble and cross-ensemble distances across all 10 ensembles.

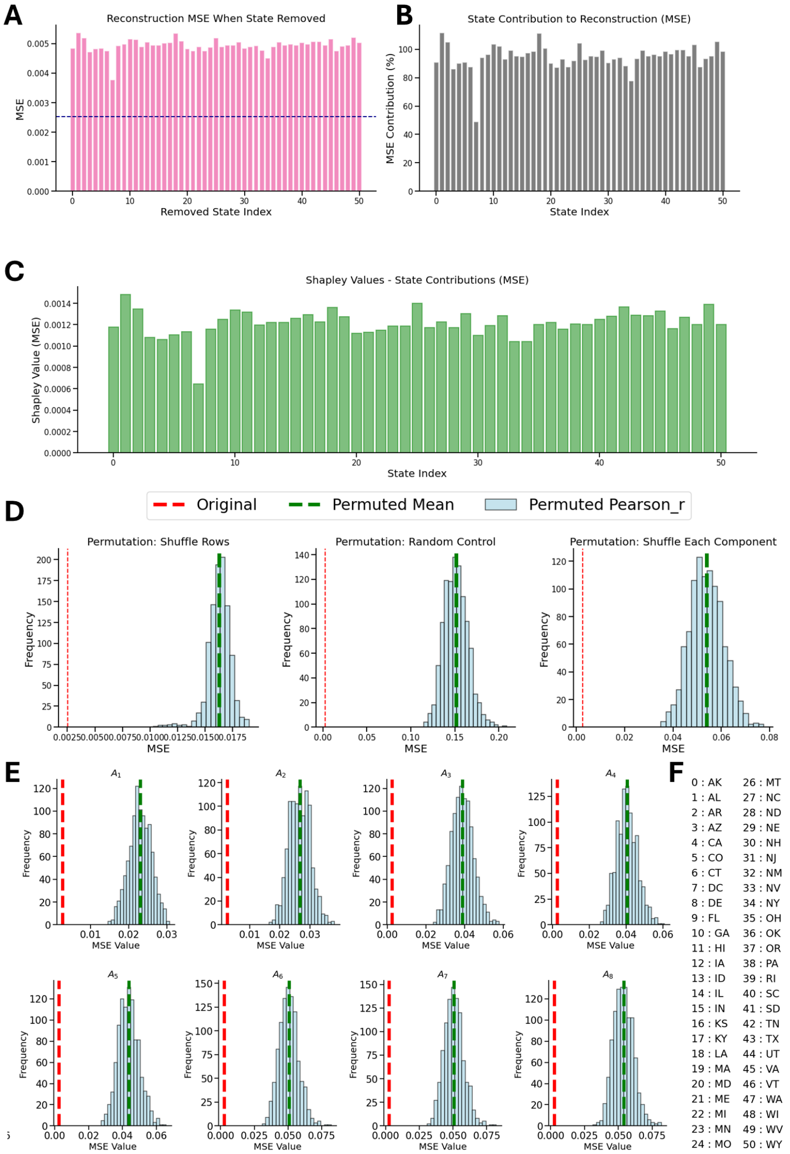


Figure 38: Post-hoc validation of MILCCI's discovered voting components (App. E.3). **A:** Leave-one-out analysis showing reconstruction MSE when each state is individually removed, with baseline MSE (dashed line) indicating model performance without omissions. **B:** Individual state contributions to reconstruction, how much each state's removal degrades performance ($100 * \frac{\text{MSE}_{\text{with omission}} - \text{MSE}}{\text{MSE}}$). **C:** Shapley values measuring each state's contribution to overall reconstruction. **D:** Permutation tests comparing original reconstruction error (red line) against three null hypotheses: shuffling state assignments between rows (left), replacing data with random noise (middle), and shuffling states within each component dimension (right). All tests show p < 0.001. **E:** Per-component permutation results demonstrating that discovered components are individually robust. **F:** State abbreviation key.

K Ethics Statement and LLM Usage

Our work does not raise any ethical concerns. Large language models were used only at the word or sentence level during manuscript writing to improve the language and catch grammar mistakes, with no influence on the scientific content or analysis.