NEUROSCIENCE

The neuronal implementation of representational geometry in primate prefrontal cortex

Xiao-Xiong Lin[1,2], Andreas Nieder[3], Simon N. Jacob[1]*

Modern neuroscience has seen the rise of a population-doctrine that represents cognitive variables using geometrical structures in activity space. Representational geometry does not, however, account for how individual neurons implement these representations. Leveraging the principle of sparse coding, we present a framework to dissect representational geometry into biologically interpretable components that retain links to single neurons. Applied to extracellular recordings from the primate prefrontal cortex in a working memory task with interference, the identified components revealed disentangled and sequential memory representations including the recovery of memory content after distraction, signals hidden to conventional analyses. Each component was contributed by small subpopulations of neurons with distinct spiking properties and response dynamics. Modeling showed that such sparse implementations are supported by recurrently connected circuits as in prefrontal cortex. The perspective of neuronal implementation links representational geometries to their cellular constituents, providing mechanistic insights into how neural systems encode and process information.

INTRODUCTION

For decades, the dominant approach to understanding neural systems has been to characterize the role and contributions of individual neurons. In a recent paradigm shift, the concept of high-dimensional activity spaces that represent cognitive and other variables at the level of neuronal populations has taken the center stage and sidelined the single-neuron perspective (*1*–*3*). These population representations capture multineuron activity in different behavioral task conditions in the form of geometrical structures (*4*, *5*). Representational geometry provides a complete description of the information encoded by and processed in a neuronal population. It does not, however, account for how individual neurons, the nuts and bolts of brain processing, give rise to the representations and the operations performed on them (*6*) because there is no direct connection between informational representation and biological implementation at the cellular and circuit level.

In constructing representational geometries, the choice of coordinate system, that is, the set of components that capture the population activity, is arbitrary. The question of what the most meaningful coordinate system is to represent the data then arises. In principal components analysis (PCA), a widely used method for dimensionality reduction, the principal components (PCs) capture the neuronal activity's variance, but they are not designed to yield biologically interpretable aspects of the representational geometry. Identifying coordinate systems that are rooted in biology is particularly relevant in association cortices where neurons often have mixed-selective responses that are not easily interpreted as the representation of any single stimulus or task variable alone (*4*, *7*). Neuronal signals in association cortices also show complex temporal dynamics and task-dependent modulations that reflect distinct sensory and memory processing stages (*8*–*10*). During working memory, for example, behaviorally relevant target items are maintained in online storage and must be protected against interfering distractors (*9*, *10*). However, depending on which coordinate system is used to express the representational geometry, the same task-related neuronal activity could be interpreted in one of two ways: either as components representing the target in each task epoch individually, suggesting a memory mechanism built on sequential relay of target information among components (*11*), or, alternatively, as components that represent the target across task epochs, suggesting a memory mechanism of continuous representation of target information by the same components (*12*).

The biological implementation of representations points to how components are accessed and information is communicated. Unlike the units in neuronal network models, in vivo neurons are subject to anatomical and physiological constraints. There are approximately 10^{10} neurons in the human brain and 10^9 in a hypothetical functional module such as the dorsolateral prefrontal cortex (PFC) (*13*, *14*). A pyramidal cortical neuron has on the order of 10^4 dendritic spines (*15*). Thus, given the disproportion between the low number of possible connections and the large number of potentially informative neurons, a neuron downstream of the PFC can only "read out" from a small fraction of neurons in this region. That is, it cannot access arbitrary components of the representational geometry. Instead, it would be more efficient and biologically plausible to read out components that a few neurons predominantly contribute to, that is, the components with a sparse neuronal implementation.

Here, we present a framework that exploits the structure in the representational geometry's neuronal implementation. We show that this approach yields components of population activity that retain links to individual neurons and does not rely on assumptions about specific neuronal activity patterns. We first tested for sparse structure in the neuronal implementation and then performed data dimensionality reduction on extracellular multichannel recordings from the nonhuman primate PFC by leveraging sparsity constraints to identify components that are contributed mainly by small subpopulations of strongly coding neurons [sparse component analysis (SCA)] (*16*, *17*). We found that the activities on these components nontrivially matched the working memory task sequence

[1]Translational Neurotechnology Laboratory, Department of Neurosurgery, Klinikum rechts der Isar, Technical University of Munich, Germany. [2]Graduate School of Systemic Neurosciences, Ludwig-Maximilians-University Munich, Germany. [3]Animal Physiology, University of Tübingen, Germany.
*Corresponding author. Email: simon.jacob@tum.de

performed by the animals, revealing separate sensory and memory components including a previously hidden component, namely, the recovery of memory content after distraction. Notably, each component was made up of nonoverlapping subpopulations of neurons with distinct spiking properties and temporal dynamics. Last, neuronal network modeling showed that recurrent connectivity as in the PFC favors such sparse implementations over nonstructured Gaussian implementations. The framework and findings presented here bridge the gap between the single-neuron doctrine and the neuronal population doctrine (*1*, *2*) and establish the perspective of neuronal implementation as an important complement to representational geometry.

RESULTS

Different neuronal implementations of the same representational geometry

Representational geometry abstracts the information coded by a population of neurons from their individual tuning profiles (*6*). It specifies the pairwise distances between task-related collective neuronal responses but no longer reflects the exact pattern of firing rates. This approach defines a stimulus-representing subspace. To illustrate, we simulated representations for two stimuli A and B in PC space, which separate, rotate, and collapse back to the origin (Fig. 1A).

The same stimulus-representing subspace can be defined with arbitrary sets of components. Components can be chosen to capture specific aspects of the representation, e.g., to continuously distinguish between stimuli (Fig. 1B) or to distinguish between stimuli in different epochs (Fig. 1C). Note that in the former example, the components align with the PCs, while in the latter, they do not. Various studies have followed this approach, selecting the components, e.g., such that they express representations sequentially (*18*) or such that they each correspond to a particular task variable of interest (*19*, *20*).

Neuronal activity can be reconstructed by the weighted sum of components. Every neuron has a set of weights quantifying its relation to the different components, i.e., its loadings on the components. The loadings of neurons on the PCs visualize their positions in implementation space (Fig. 1, D to F), where the loadings along any axis correspond to a component in representation space with the same orientation (Fig. 1, A to C). The structure in the implementation space, i.e., the distribution of loadings across neurons, can be exploited to identify a unique, nonarbitrary set of

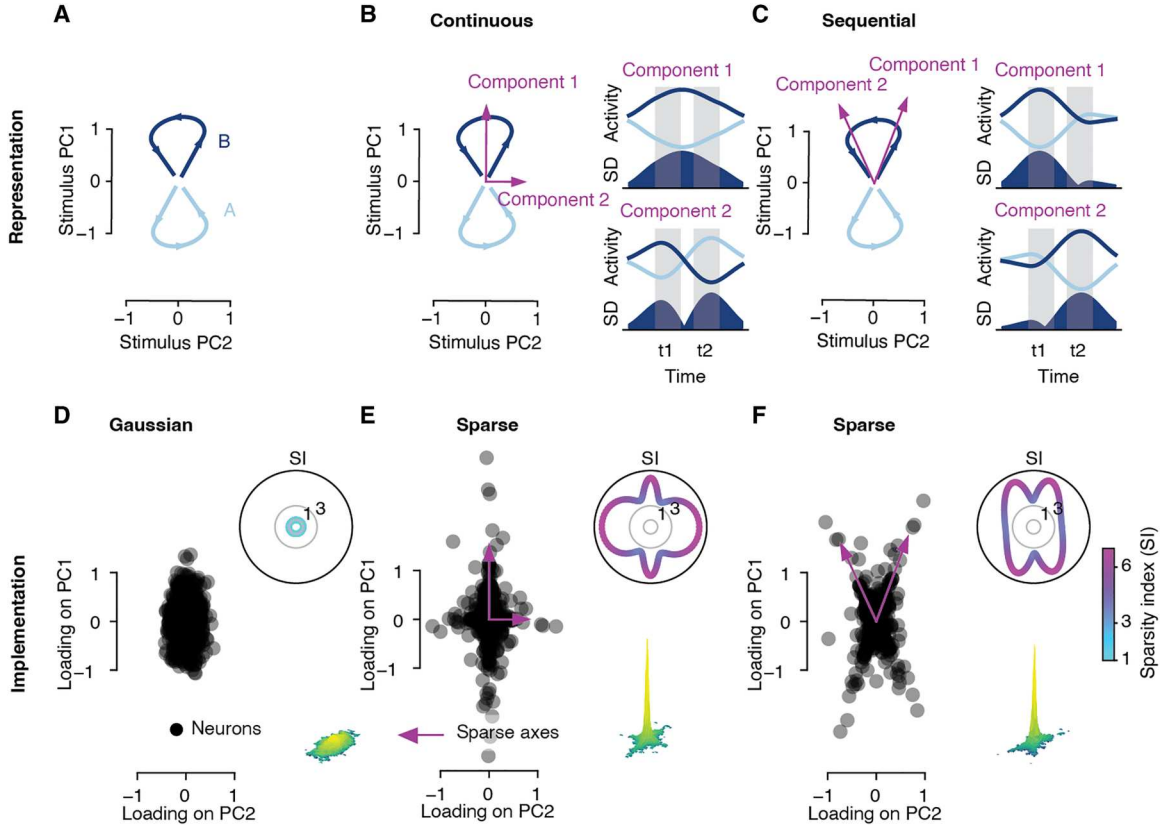

Fig. 1. Different neuronal implementations of the same representational geometry. (**A**) Representational geometry for two trials with stimuli A and B on the plane specified by stimulus PC1 and PC2. Time runs along the individual trajectories. (**B**) Left: Example pair of components that express the representational geometry (magenta arrows). Right: Activities on the corresponding components and standard deviation (SD) across components as a measure of amount of information carried by them. Both components represent stimulus information in both epochs. (**C**) Same layout as in (B) for components that each represent stimulus information in one epoch only. (**D** to **F**) Neuronal implementation underlying the representational geometry in (A to C), specified by the distribution of neuronal loadings on the stimulus PCs. Insets: SI of all axis orientations in the space spanned by PC1 and PC2. Axes with high SI (sparse axes, magenta arrows) in (E) and (F) correspond to the components 1 and 2 in (B) and (C), respectively.

components that relies on priors about anatomical connectivity instead of on priors about activity patterns.

Representational geometry is invariant to the rotation of neuronal coordinates (*21*). Different neuronal implementations may therefore underlie the same representational geometry. We first consider the scenario of a nonsparse (Gaussian) distribution of loadings (Fig. 1D), where the standardized moments (e.g., skewness and kurtosis) are constant, meaning there are no differences in these distributional statistics across axis orientations. We define the sparsity index (SI; Fig. 1D, top inset) to denote the sparsity of the implementation along a given axis. SI is proportional to a distribution's kurtosis and is defined as 1 for a Gaussian distribution. If SI is constant across axis orientations, then neurons do not preferentially align to any axes.

Next, we consider a sparse distribution (Fig. 1E). Most neurons lie around the origin of the coordinate system. However, because SI is not constant (Fig. 1E, top inset), we can find the sparse components (SCs) that strongly coding neurons align to. In the present case, these sparse axes correspond to the components in representational space that code the difference between stimulus A and B continuously (with one of the components reversing between epochs; compare Fig. 1E with Fig. 1B). Sparse distributions can exist for arbitrary axis orientations. For example, strongly coding neurons could align to the components that sequentially represent the stimulus information in epoch 1 and epoch 2 (compare Fig. 1F with Fig. 1C).

Although both scenarios are characterized by sparse neuronal implementations, we note that they have fundamentally different implications for readout, lending particular importance to the positioning of sparse axis orientations. Continuous readout [Fig. 1, B and E (component 1)] is stable but not optimized for either epoch 1 or epoch 2, whereas sequential readouts (Fig. 1, C and F) are more precise at the respective epochs, but not stable across epochs. In summary, the perspective of neuronal implementation offers a way to connect representational geometries to their cellular constituents, revealing mechanistic insights into how a neural system encodes, processes, and relays information.

The neuronal implementation of working memory

With this framework, we now examine neuronal implementation of working memory, a core cognitive function for online maintenance and manipulation of information in the absence of sensory inputs. Extracellular multichannel recordings were performed in the lateral PFC of two monkeys trained on a delayed-match-to-numerosity task, requiring them to memorize the number of dots (i.e., numerosity) in a visually presented sample and resist an interfering distracting numerosity (Fig. 2A) (*10*). A total of 467 single units recorded across 78 sessions were included in the analysis. Spike rates were binned, averaged across conditions of the same type, and demixed into their constituent parts (Fig. 2B) (*22*). Because the task design was balanced (i.e., all sample-distractor combinations were included), the different task variables were statistically independent of each other. Demixing therefore allowed to isolate and analyze signal components that would otherwise be overshadowed by signals that dominate the raw firing rates. Across neurons, the neuronal activities coding for trial time, sample numerosity, distractor numerosity, and the sample-distractor interaction accounted for 72.7, 8.7, 5.8, and 12.9% of the total variance, respectively (Fig. 2B).

We first focused on the representation of the sample numerosity throughout the trial, the crucial function for completing the task (Fig. 2C). In PC space, the representations of different numerosities (1 and 4 visualized here) started to separate, marking an increase of the information during sample presentation. Then, the representations rotated and returned to the origin. Similar representational changes have been reported previously (*11*, *23*, *24*).

The distribution of loadings of individual neurons onto the first three PCs was highly non-Gaussian ($P < 0.001$; Henze-Zirkler multivariate normality test; Fig. 2D). Accordingly, the SI was not uniform across all axis orientations (Fig. 2D, inset). The sparsity parameter β fit to neuronal loadings on randomly chosen axes was significantly smaller than the β of a Gaussian distribution, indicating sparser distributions ($P < 0.001$; bootstrap; Fig. 2E). These analyses confirmed the presence of sparse structure in the neuronal implementation.

Using SCA that identifies components with sparse distributions of neuronal loadings (SCs), we found three SCs that optimally decomposed the sample numerosities' representational geometry (i.e., the minimum number of components that retained 95% of the maximal explained variance). The SCs displayed temporally well-defined active periods that matched the task structure and tiled the duration of a trial (Fig. 2F). Intuitively, they correspond to components for sensory encoding, memory maintenance, and memory recovery following distraction, in accord with the scenario of sequential representations [compare to Fig. 1 (C and F)].

To control for the possibility that noise in nonsparse implementations is mistaken for structure by SCA, we created substitute datasets with random Gaussian implementations (i.e., Gaussian distributions of neuronal loadings) while keeping the representational geometry intact and then systematically compared the original SCs with the substitute SCs (example substitute SCs in Fig. 2G). First, the loadings on the three sparse axes in the original data were sparser than the loadings on the sparse axes in the Gaussian substitutes ($P < 0.001$ for all three sparse axes; permutation test with $n = 3 \times 1000$ permutations; Fig. 2H). Second, compared to the substitutes, the SCs in the original data showed temporally restricted sample representations with shorter spread [$P = 0.002$; permutation test with $n = 1000$ permutations; same as for Fig. 2 (I to K); Fig. 2I], less temporal overlap with each other ($P = 0.003$; Fig. 2J), and less reversal of sample numerosity tuning ($P = 0.030$; Fig. 2K), suggesting that the observed SCs' activity was more sequential than that of the SCs obtained from substitutes with a random Gaussian implementation. Third and last, the SCs were closer to orthogonal than the substitutes ($P = 0.019$; Fig. 2L), demonstrating that the observed implementation is more efficient than a random implementation.

In addition, to verify that the sequential representation was not an artifact of SCA, we constructed three continuously active components from the original representational geometry and then created synthetic neurons such that the distribution of their loadings on these components were sparse. SCA was able to recover those continuous components and did not produce sequential representations (fig. S1).

In summary, the neuronal implementation of the sample numerosities' representational geometry was structured and sparse. The activities on the SCs demonstrated sequential rather than continuous coding of working memory content, indicating that the change of behavioral demands in the course of the trial triggers a switching of informative subpopulations.

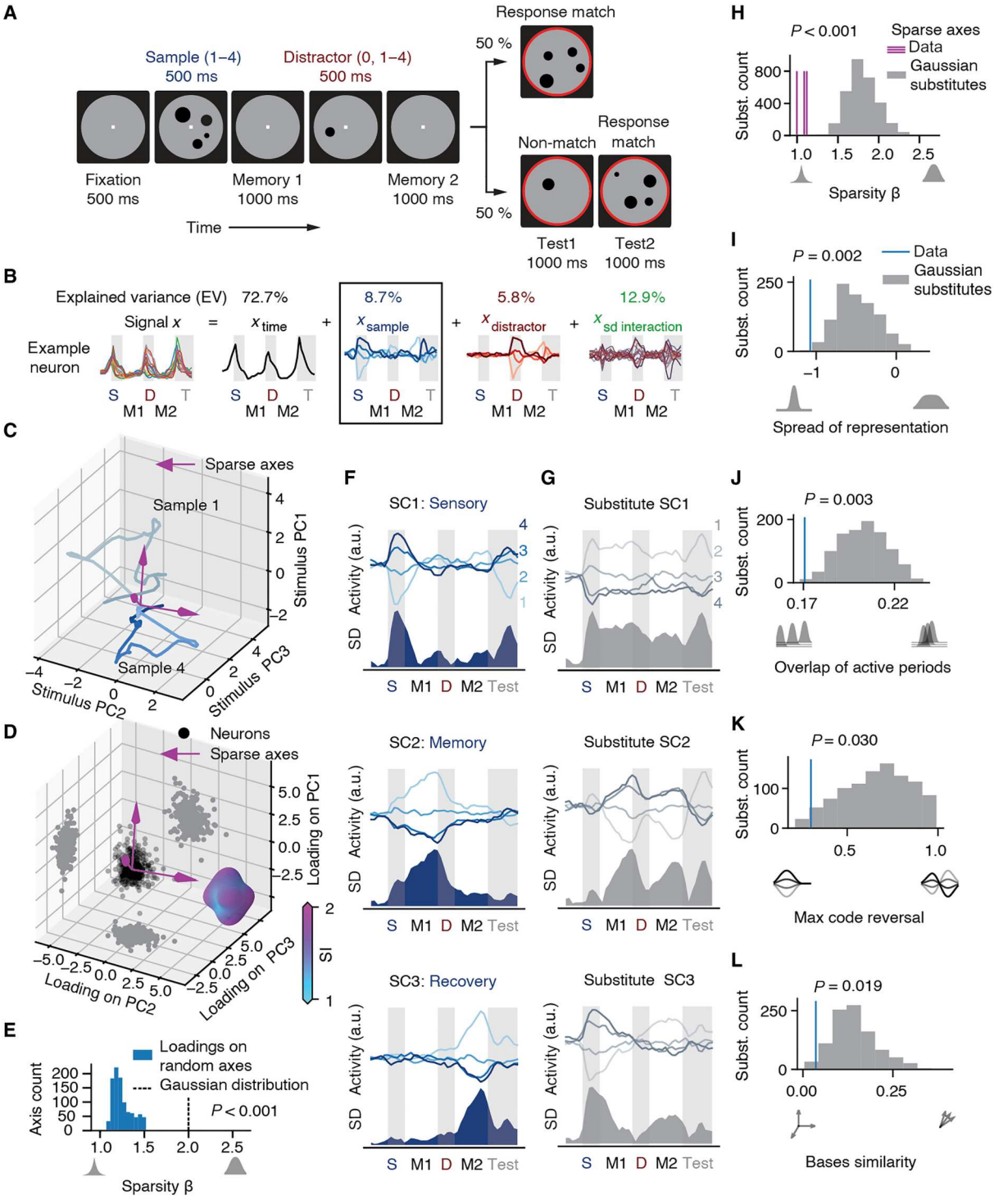

Fig. 2. The neuronal implementation of working memory. (**A**) Delayed-match-to-numerosity task with distractors. (**B**) Demixing procedure separating the activity of each neuron into the parts coding time, sample numerosity, distractor numerosity, and sample-distractor interaction. The sample coding part is used for the following analyses. Top: Percentage of explained variance for each part. (**C**) Representational geometry for sample numerosities 1 and 4 in PC space, averaged across trials of the same condition. (**D**) Loadings of all recorded neurons on the top three PCs (black dots) including distributions projected onto the planes formed by PC pairs (gray dots). Sparse axes (magenta arrows; determined by SCA) have high SI. Inset: Surface plot of SI for all axes in the space. (**E**) Sparsity parameter β for neuronal loadings on randomly selected axes (n = 1000 bootstraps) compared to the β of a Gaussian distribution. (**F**) Activity of the three identified SCs, averaged across trials for each sample numerosity condition (top; numbers indicate sample numerosity) and relative information across conditions measured as SD. (**G**) SCs of an example substitute dataset with nonsparse Gaussian implementation. (**H**) Sparsity β of the neuronal loadings on the SCs for the original data and the substitute datasets (permutation test with $n = 3 \times 1000$ permutations). (**I** to **L**) Activity measures for the SCs of the original data and the substitute datasets (permutation test with n = 1000 permutations). a.u., arbitrary units.

The effect of distraction on sample numerosity representations

The lack of an SC that continuously represented the behaviorally relevant sample numerosity throughout the trial was intriguing, tapping into the question of stable versus dynamic memory coding (*11*, *12*, *25*–*27*). We therefore investigated the influence of distraction on sample number coding in more detail.

First, we applied SCA to the demixed distractor coding part of the data (Fig. 3A, top). Two SCs were obtained that were sequentially active during presentation and maintenance of the distractor numerosity, respectively (Fig. 3A, bottom). These components resembled the sensory and memory sample coding SCs (compare to Fig. 2F), suggesting that target and distracting information initially occupied similar resources despite their distinct behavioral relevance. Supporting this hypothesis, we found strongly overlapping

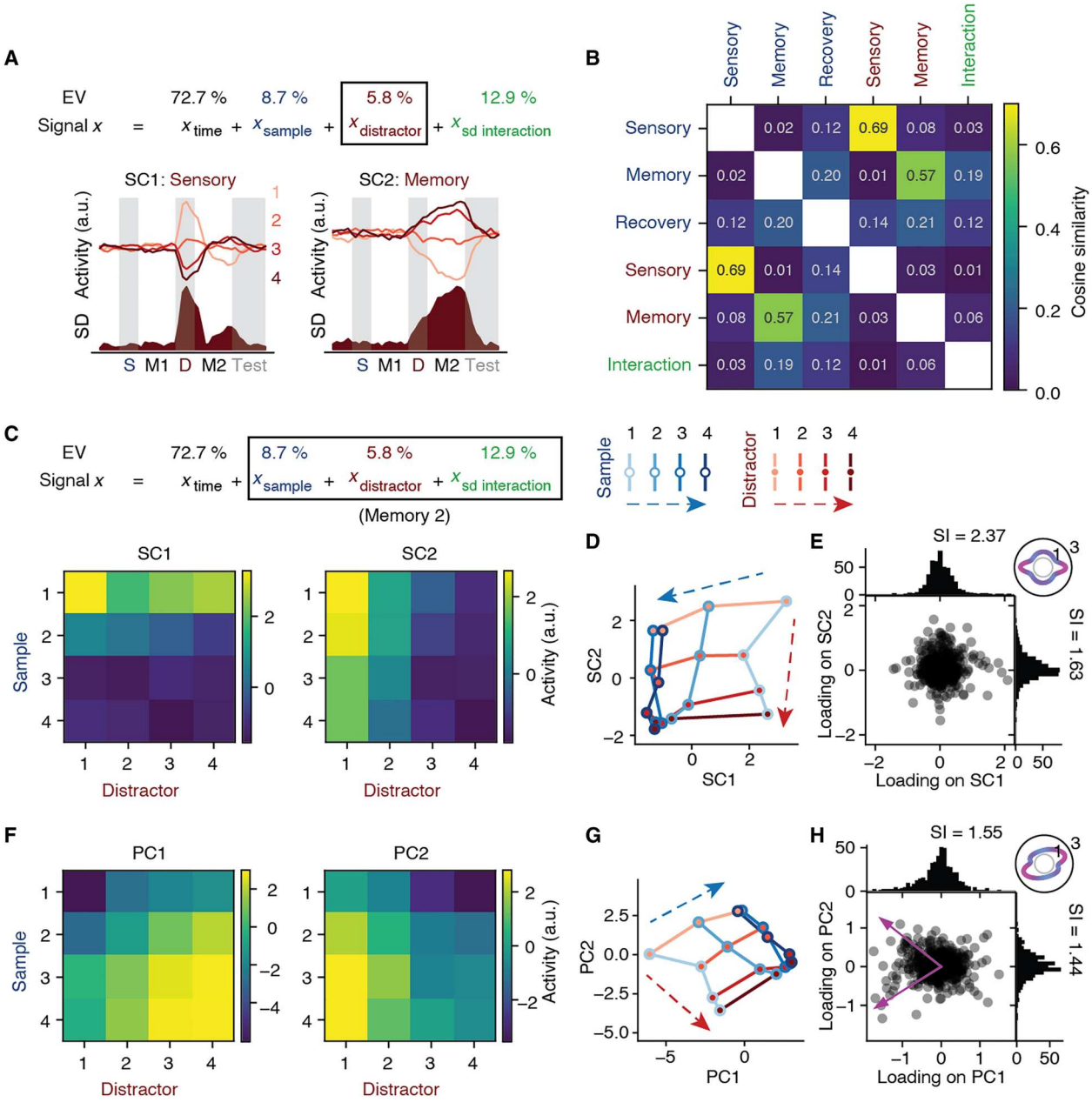

Fig. 3. The effect of distraction on sample representations. (**A**) Top: The demixed distractor representing part used in the analysis. Bottom: Distractor numerosity SCs. Numbers indicate distractor numerosity. (**B**) Cosine similarity between loadings of sample numerosity SCs (blue), distractor numerosity SCs (red), and the sample-distractor interaction SC (green). (**C**) Activity of the two SCs identified using firing rates averaged across the second memory delay for all sample-distractor combinations without demixing the stimulus presentations. (**D**) Representational geometry in SC space. Blue and red colors indicate sample and distractor numerosity, respectively. The blue and red arrows visualize the sample and distractor coding axes, respectively. (**E**) Neuronal loadings on the two SCs. Dots, joint distribution in SC space; histograms, marginal distribution of neuronal loadings on SC1 and SC2. Inset: SI for all axes. (**F** to **H**) Same layout as in (C to E) but for PCs. Magenta arrows in (H) indicate sparse axes.

neuronal loadings between sample SCs and distractor SCs (cosine similarity; 0.69 and 0.57 for the sensory and memory components, respectively; Fig. 3B) with displacement of sample information by distractor information as the trial evolved (fig. S2A, top and middle). However, in contrast to the sample sensory and memory components, the sample recovery SC was unique and did not share loadings with any other SC (Fig. 3B). Furthermore, the sample recovery SC was not influenced by distractor information and carried sample information until test numerosity presentation (fig. S2A, bottom). More activity in the sample sensory and recovery SCs was observed in trials with a distractor than in trials without a distractor (fig. S2B). Conversely, sample information was lower in the memory component when a distractor was presented.

Second, we applied SCA to the sample-distractor interaction part of the data. One SC was identified. Its activity was most pronounced when the sample and distractor numerosity were the same (fig. S2C). The neuronal loadings on this SC did not overlap with the loadings on sample or distractor SCs (Fig. 3B), suggesting that the boost in numerosity information was generated by a dedicated subpopulation responding to a repeated presentation of the same number, instead of changing the activity of the sample representing neurons.

Together, these results indicate a (partially) shared capacity for sample and distractor representations during the sensory input and subsequent memory delay stages. The invasion of distractor information necessitated the recruitment of an extra component, the recovery component, to maintain sample information in working memory. This also occurred on trials without a distractor (fig. S2B). We speculate that this was because our task design included 80% of trials with a distractor. In other words, the animals were extensively trained to expect and handle memory interference. The extent of recovery could be different in a task design with more balanced distractor trials.

So far, all analyses were performed on separated (demixed) representations. We next investigated whether sample and distractor information could be equally disentangled using SCA alone without demixing the numerosity coding signal (Fig. 3C). SCA performed on firing rates averaged across the second memory delay recovered two SCs that each selectively captured sample and distractor information (Fig. 3C). The corresponding representational geometry was grid-like. The grid was nonuniformly spaced, reflecting the size effect in analog magnitude estimation (*28*). The sample coding axes had similar orientations for each distractor and vice versa (Fig. 3D), reflecting factorized sample and distractor representations. Notably, the sample coding axes and distractor coding axes aligned well to the corresponding SC. This was not enforced by our analytical method, arguing that the PFC spontaneously disentangles target and distractor representations in working memory. The underlying implementation showed clear sparse structure in the neuronal loadings onto these components (Fig. 3E; SI).

For comparison, PCA, which is insensitive to the neuronal implementation, was unable to recover factorized components (Fig. 3F). The grid-like geometry was still largely preserved, but it did not align with the PCs (Fig. 3G). In contrast to SCA, PCA did not identify the components with the sparsest loadings (Fig. 3H).

Subpopulations of neurons dominating working memory representations

Next, we investigated whether the implementation was sparse enough to be able to reliably reconstruct the population-level sample representation using only a small fraction of neurons. We performed cross-temporal linear discriminant analysis (LDA) to decode sample numerosity at a given time point in the trial using training data from a different time point (Fig. 4). Decoding accuracy therefore quantifies the degree to which the representation is transferable. With four numerosities, chance level accuracy is 25%. Using the entire population of 467 recorded neurons, we found a dynamic code for the sample numerosity with good within-epoch transfer but very little generalization across epochs, particularly from the first to the second memory delay (Fig. 4A). In line with our previous results, this finding suggests that working memory representations are nonuniform and that distinct, complementary processes are required to protect behaviorally relevant information from interference.

We selected the neurons that contributed most to the previously identified SCs (loading on the SC larger than two SDs; Fig. 4B). Thirty-six, 28, and 28 single neurons passed the criterion for the sensory, memory, and recovery SC, respectively. Although each subpopulation composed only 6 to 8% of the entire recorded population, these dominant neurons explained 88, 82, and 87% of their respective component′s variance (sum of squares of dominant neurons′ loadings over sum of squares of all neurons′ loadings). Overlapping membership in two subpopulations was very rare (no more than three neurons in any SC pair; Fig. 4B).

Cross-temporal LDA using only the dominant neurons showed a very similar sample numerosity decoding pattern as with the entire population (Fig. 4C, compare to Fig. 4A), confirming that the decoder previously relied mainly on this small subset of neurons. When all the dominant neurons were removed, the sample numerosity neuronal representation did not generalize within epochs, instead changing rapidly across time (fig. S3A). This suggested that the dominant neurons were responsible for the stable sample representation within each task epoch. The sensory subpopulation contributed to decoding in particular during the sample and test numerosity presentation but showed very little activity in the memory epochs (Fig. 4D, top). The memory subpopulation dominated in the first delay but unexpectedly was not involved in sample coding during the second delay (Fig. 4D, middle). Instead, after distraction, the recovery subpopulation was exclusively responsible for carrying sample information (Fig. 4D, bottom). This suggests that these neurons crucially contribute to shielding working memory information from interference (see also fig. S2).

Distractor information could also be decoded from the population (fig. S3B). Successful cross-stimulus decoding (training on sample numerosity and testing on distractor numerosity; fig. S3C) implied that the neurons that represented sample numerosity in the sample epoch and in the first memory epoch turned to represent the distractor in the same fashion in later epochs, consistent with our previous result of overlapping neuronal loadings for distractor and sample components with the notable exception of the sample recovery component (Fig. 3B).

Last, to further validate our results, we examined specifically whether a stable representation of sample information across both memory delays could be implemented by a unique neuronal subpopulation. No single neurons had high loadings on both the memory

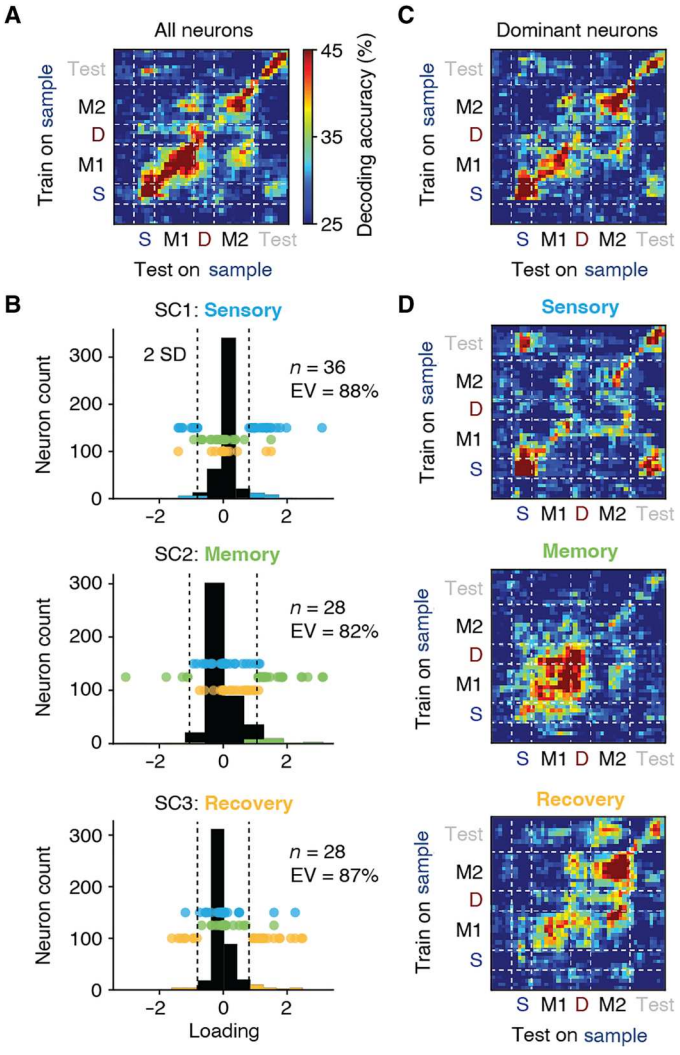

Fig. 4. Subpopulations of neurons dominating working memory coding. (**A**) Accuracy of cross-temporal LDA decoding of sample numerosity using all recorded neurons. (**B**) Neuronal loadings on the three identified sample numerosity SCs. Colored dots indicate the "dominant" neurons selected in each SC (cutoff: 2 SD). The percentage of variance explained within each SC is given for each subpopulation. (**C**) Accuracy of cross-temporal LDA decoding of sample numerosity using only the dominant neurons. Compare to (A). (**D**) Sample numerosity decoding accuracy using the dominant subpopulations of each SC. Same color scale in (A), (C), and (D).

and recovery components (fig. S4A; no data points in the top right and bottom left corner). This was also reflected in the low SI of the loading distribution on the diagonal (fig. S4A, inset). Accordingly, LDA decoders trained to decode sample information continuously across both memory epochs relied on more neurons (58 neurons) than the decoders trained in the first memory delay (23 neurons) or in the second memory delay (36 neurons; fig. S4B). This result suggests that a continuous representation of sample information across epochs was effectively implemented by the summation of two sequential components.

Subpopulation-specific spiking properties

Above, we identified dominant neurons based on their stimulus selectivity. We now investigated whether their different roles in representing sample information were possibly mirrored by distinct spiking properties.

First, we calculated the across-trial similarity (Pearson correlation) between each neuron's activity at different time points in the fixation period to derive the intrinsic time scale, a measure considered to index a neuron's ability to maintain memory traces (*29*). Representative neurons from all three subpopulations are shown (Fig. 5A). The example recovery neuron had a substantially larger spread from the diagonal than the sensory and memory neuron, i.e., its activity in distant time points was more strongly correlated, thus signifying a longer time constant (Fig. 5A, bottom). For each subpopulation, an exponential decay was fitted to the mean correlation coefficient across neurons (Fig. 5B). The recovery subpopulation had the largest time constant τ (165, 127, and 338 ms for sensory, memory, and recovery neurons, respectively). The distribution of τ values in the recovery population also stood out from the distributions observed in subsampled subpopulations of PFC neurons, whereas the sensory and memory neurons' distributions were not significantly different [$P = 0.874$, $P = 0.455$, and $P = 0.002$ for sensory, memory, and recovery subpopulations, respectively; Kullback-Leibler (KL) divergence with bootstraps; Fig. 5C].

Next, we investigated spike train statistics using the interspike intervals (ISIs) measured during the neurons' entire recording lifetime. The coefficient of variation (CV) measures the irregularity of a spike train (Fig. 5D). CVs of all recorded neurons were larger than 1 (i.e., more irregular than a Poisson process) with a gradual increase of spiking irregularity across the sensory, memory, and recovery subpopulations. CVs in the recovery neuron population were significantly larger than in the sensory subpopulation ($P = 0.030$, two-tailed t test; Fig. 5D), possibly reflecting their unique temporal dynamics. The local variation (LV) measures local ISI differences and complements CV, which is a global measure. LV reflects the instability of firing at a small time scale. LVs in all dominant neurons were smaller than 1 (i.e., less LV than a Poisson process) and significantly lower than in the noncoding PFC population ($P < 0.001$, two-tailed t tests; Fig. 5E), potentially underlying their ability to stably represent sample information within epochs via persistent activity.

These distinct spiking properties likely reflect the distinct local circuitry each dominant subpopulation is embedded in. Notably, these measures were not involved in the original selection of subpopulations and therefore lend support to the notion that the implementation structure carries biological meaning.

Subpopulation-specific temporal dynamics and representation of context

There was no perceptual cue in the working memory task specifying the difference between sample and distractor. This forced the animals to internally keep track of a trial's temporal evolution. To investigate whether temporal dynamics and context played a role in supporting the subpopulation-specific stimulus representations, we next analyzed the temporal part of the demixed signal.

The temporal part drove most of the variability (72.7%) of the firing rates in the recorded population and occupied a higher-dimensional space than the sample coding part (fig. S5A). Neuronal loadings on random axes within this space were sparse ($P < 0.001$;

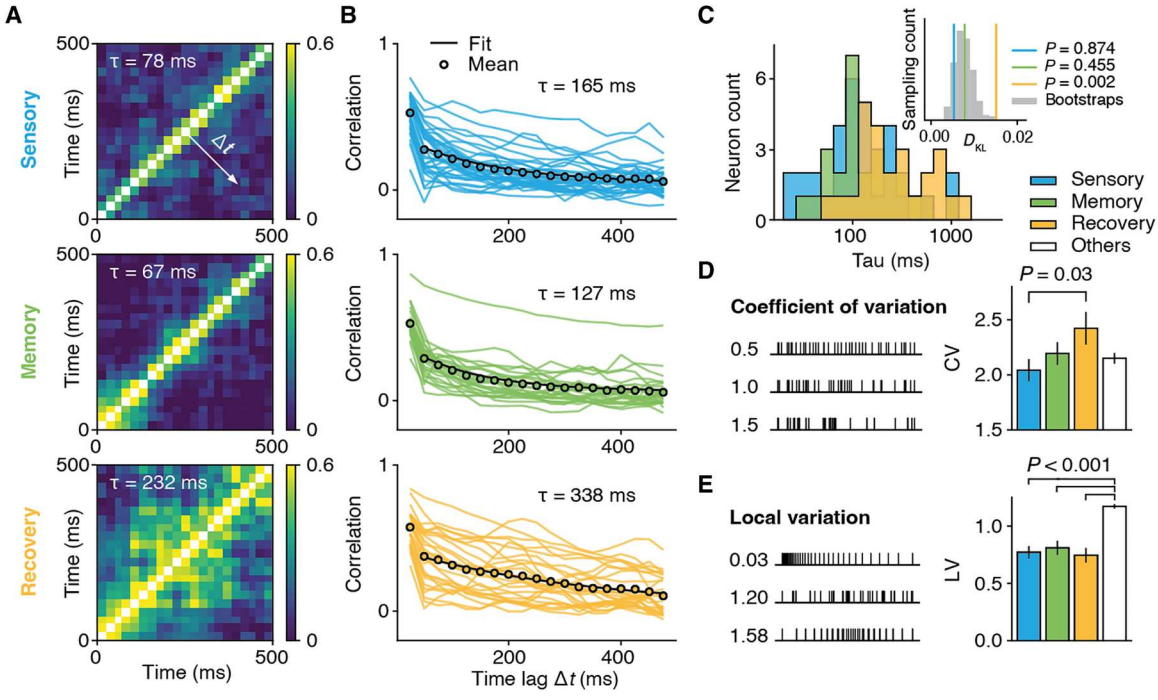

Fig. 5. Subpopulation-specific spiking properties. (**A**) Between–time point Pearson correlations of the trial-to-trial fluctuation of firing rates in the fixation epoch for the three dominant subpopulations. (**B**) Autocorrelograms obtained by averaging across diagonal offsets in (A). Autocorrelograms of individual neurons are given (single lines) together with the subpopulation average and the fitted exponential decay (black dots and line, respectively). (**C**) Distribution of fitted decay constants of individual neurons in each dominant subpopulation. Inset: Kullback-Leibler divergence (D_{KL}) between the distribution of each subpopulation and the whole population (null distribution for significance testing created with $n = 1000$ bootstraps from the whole population). (**D**) CV of ISIs of the dominant subpopulations and the nondominant other neurons (two-tailed t test). Left: Example spike trains for different CVs. (**E**) Same layout as in (D) for the LV of ISI.

bootstrap; fig. S5B). We therefore also applied SCA to the temporal part and found that the SCs exhibited a variety of distinct activity patterns modulated by the individual trial events (fig. S5C). In each of the sample coding dominant subpopulations, the temporal part formed unique trajectories (Fig. 6A). In the sensory subpopulation, the trajectory followed a periodic, quasi-circular course (Fig. 6A, top). The first and second memory epochs overlapped almost entirely. This indicates that the sensory neurons did not distinguish between the time periods after sample and after distractor presentation. The trajectory of the memory subpopulation was less periodic but intertwined in the first and second memory epochs (Fig. 6A, middle). In contrast, the trajectory of the recovery subpopulation was less intertwined, with most time points distinguishable from each other, especially the first and second memory epochs, signifying a better representation of the contextual difference following sample and distractor presentation (Fig. 6A, bottom).

Overlap of the memory epochs in the sensory and memory subpopulations could be due to the limitations of a linear projection and the emphasis of PCA on global structure. We therefore performed nonlinear embedding using t-distributed stochastic neighbor embedding (t-SNE; two-dimensional embedding; perplexity = 30; Fig. 6B). This analysis revealed comparable structures as the linear projection, with the first and second memory epochs separated only in the recovery neuron subpopulation.

To further investigate the temporal evolution of neuronal activity, we measured the Euclidean distances between individual time points in each subpopulation (full space; Fig. 6C). All distance matrices displayed a strong diagonal, reflecting the fact that close-by time points were represented similarly. Notably, there were also strong offset diagonals in the sensory subpopulation, meaning that activity in these neurons repeated with a cycle of about 1.5 s, the interval between sensory onsets (sample, distractor, and test numerosities). Furthermore, activity in the sensory and memory epochs differed the most in this subpopulation. These patterns were present, albeit weaker, in the memory subpopulation, but absent in the recovery neurons. We quantified periodicity for each neuron by computing the relative power of 1/1.5 s (0.67 Hz) activity and its harmonics normalized to the power of the full frequency spectrum (Fig. 6D). If a neuron responded in the same way to all trial events, its periodicity would be high. Conversely, if a neuron showed different responses to the different trial events, its periodicity would be low. In other words, the periodicity measure quantifies how sensitive a neuron is to differences in temporal context. Compared to randomly sampled subpopulations of PFC neurons, the sensory subpopulation and the recovery subpopulation showed significantly different (higher and lower, respectively) periodicity ($P < 0.001$, $P = 0.051$, and $P = 0.043$ for sensory, memory, and recovery subpopulations, respectively; KL divergence with bootstraps; Fig. 6D, inset).

Neuronal activity is not static and temporally independent. Instead, firing rates at every time point depend on previous time points. To characterize the dynamical properties of the recorded PFC population in more detail, we used the measure of tangling (*30*). Tangling measures the extent to which the velocity (direction and speed) of a given state on a trajectory diverges from the velocity of its neighboring states (Fig. 6E), reflecting the level of

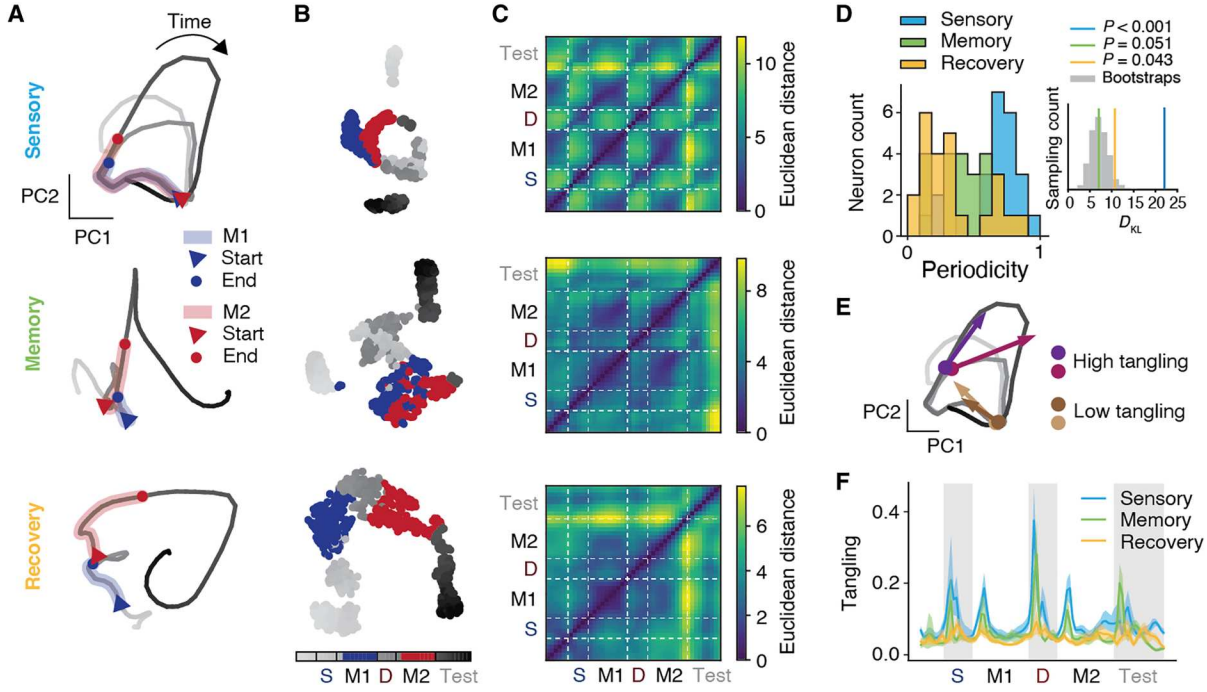

Fig. 6. Subpopulation-specific temporal dynamics. (**A**) Temporal part of the demixed neuronal activity (averaged across conditions) of each dominant subpopulation projected onto their respective top two PCs (explained variance: 52.5 and 21% in the sensory subpopulation, 39.4 and 23.5% in the memory subpopulation, and 44.3 and 22.5% in the recovery subpopulation). Time runs along the individual trajectories (bin width, 50 ms). First and second memory delays are marked in blue and red, respectively. (**B**) Full signal averaged within each condition and embedded in 2D *t*-SNE space. Bins as in (A). (**C**) Euclidean distances between time points on the trajectory in (A) of each subpopulation. (**D**) Distribution of periodicity (relative power of 1/1.5 Hz and harmonics) of individual neurons in each subpopulation. Inset: D_{KL} between the distribution of each subpopulation and the whole population (null distribution for significance testing created with $n = 1000$ bootstraps from the whole population). (**E**) Example time points on the trajectory of the sensory subpopulation with high and low tangling. (**F**) Time-resolved tangling of the trajectory of each subpopulation.

unpredictability and instability (chaos) in the system. High tangling means that a small disturbance in the current state would lead to large changes in the next state (difference of derivatives of neighboring points). The instability or inability to determine the next state from the current state (i.e., high tangling) indicates that other neuronal populations or external stimuli may drive the trajectory. Consequently, tangling was increased following the onset and offset of sensory input in all three subpopulations. Tangling was highest, however, in the sensory subpopulation and lowest in the recovery subpopulation (sensory versus memory, $P < 0.001$; memory versus recovery, $P = 0.013$; two-tailed t test across all trial time points; Fig. 6F).

In summary, these results suggest that the subpopulation of recovery neurons keeps a record of time and temporal context, which could contribute to these neurons′ ability to separate sample and distractor information. In contrast, the sensory subpopulation, as well as the memory subpopulation to a lesser degree, is characterized by its strong input-driven temporal dynamics, which is consistent with these neurons′ passive representation of numerosity regardless of it being behaviorally relevant (sample) or irrelevant (distractor).

Sparse implementations favored in recurrent circuits

The decomposition we used is mathematically equivalent to a feed-forward neural network (linear autoencoder with one hidden layer) (*31*). In the feed-forward case, sparse connections directly lead to sparse neuronal loadings. With the addition of more layers, the neuronal loadings could finally approximate a nonsparse (Gaussian) distribution. The PFC is a highly recurrent, rather than purely feed-forward, brain region (*32*). The recurrent connections could, in effect, function as multiple feed-forward layers, potentially allowing a neural network with sparse connections to produce nonsparse (Gaussian) neuronal loadings equally well. To address this, we created synthetic datasets with different sparsity of neuronal loadings and tested whether a recurrent neural network (RNN) model would still favor sparse implementations.

The RNN model was trained to reproduce the target (to-be-fitted) neuronal firing rate sequences of each sample-distractor combination (Fig. 7A). The model consists of 467 neurons (to match the recorded population) receiving inputs of stimulus information according to the task structure. The model learns the recurrent connectivity W among the neurons. W summarizes the influence of the current time point's firing rates $\mathbf{r}$ on the firing rates of the next time point. An indicator vector $\mathbf{n}$ (one nonzero entry) represents the sample and distractor numerosity, activating the numerosity-specific input in I to the entire neuronal population. To reflect the absence of an explicit visual cue that differentiates between sample and distractor in the task design, sample and distractor numerosity share the same input channel I. The contextual difference is left for the model to resolve. The intercept term $\mathbf{b}$ captures the baseline activity of each neuron.

We first trained the model on the original dataset and visualized the trajectory of the output averaged across all conditions (Fig. 7B). The model reproduced the original dataset well, capturing 85.7% of

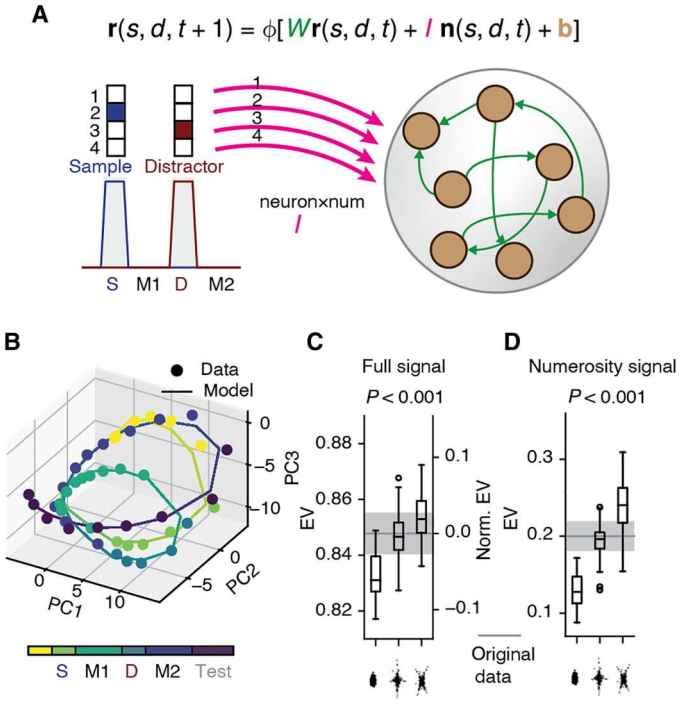

Fig. 7. RNN modeling. (**A**) RNN model governing equation and structure. Magenta and green arrows indicate numerosity-specific inputs and connectivity weights to be trained, respectively. (**B**) Model fit (solid trajectory) to original data (dots) averaged across all conditions. (**C**) Percentage of variance of the full signal explained by the model for nonstructured Gaussian implementations of numerosity representations (left bar), sparse implementations with random orientations of sparse axes (middle bar), and sparse implementations with the same orientation of sparse axes as in the original data (right bar). Left and right axis show explained variance relative to the full signal and to the manipulated signal, respectively (one-way ANOVA across substitutes). (**D**) Same layout as in (C) for the percentage of variance of the numerosity signal explained by the model.

total variance. Next, we created substitute datasets with altered implementations of numerosity representations ($x_{\text{sample}} + x_{\text{distractor}} + x_{\text{sd interaction}}$) for the model to fit. The temporal part of the demixed data was unchanged. Three different implementations were created: first, a nonsparse Gaussian distribution of neuronal loadings and no alignment to any components (compare to Fig. 1D), second, a distribution with the same degree of sparsity as the original data but with sparse axes randomly rotated to align to other components (compare to Fig. 1E), and third, a substitute with the same sparse distribution of neuronal loadings as in the original data (compare to Fig. 1F).

The model captured an increasing proportion of variance of the full signal across the three substitutes [$P < 0.001$; one-way analysis of variance (ANOVA); Fig. 7C]. The absolute differences in explained variance were comparatively small (left axis) but remarkable in relation to the variance of the manipulated signal (right axis) and given that the representational geometry was unchanged and identical for all substitutes (compare to Fig. 1). A comparable result was obtained for the explained variance of the numerosity coding part ($P < 0.001$; one-way ANOVA; Fig. 7D). Together, these results demonstrate that sparse implementations of working memory representations are favored by neural networks with recurrent circuits, the characteristic wiring motif of association cortices such as the PFC.

DISCUSSION

We presented a framework to examine the contributions of individual neurons to population-level responses in representation space and to utilize its implementation structure. We identified heavy-tailed (i.e., sparse) distributions of neuronal loadings on components that captured disentangled and sequential memory representations including the recovery of memory content after distraction. The switching of working memory components circumvented interference. These components could be traced to small subpopulations of neurons with distinct spiking properties and temporal dynamics. Modeling showed that such sparse implementations with sequentially active components are supported by recurrently connected networks.

Bridging population activity and neuronal implementation

Population-level activity and representational geometry were previously studied without forming direct links to individual neurons (*4*–*6*, *33*). However, while single-neuron selectivity measures have the advantage of being more easily connected to biological properties such as cell type, receptor expression and axonal projection targets, they are typically chosen based on intuition and past experience and only partially or indirectly reflect the full representational space (*10*, *34*).

Our SCA framework (Fig. 1) combines the advantages of both perspectives. It builds on representational geometry for a comprehensive account of the data and then links the relevant coding dimensions in the activity space to populations of strongly contributing neurons, which allows relating the population-wide activity patterns to tangible physiological measures.

Capturing biologically meaningful dimensions in activity space

SCA examines neuronal implementation without the need to manually construct the components to which individual neurons contribute (*4*, *19*). SCA is not limited regarding the number of selectivity indexes to examine at a time; it does not require temporal averaging, allowing us to investigate datasets with rich temporal modulation and higher-dimensional stimulus coding; it does not overweigh neurons with low selectivity, effectively discarding noise. Conversely, sparse loadings can be overlooked by selecting a limited set of selectivity indexes and averaging across time (*4*), which is equivalent to examining the weighted sums of the independent sources. This can introduce a bias in neuronal loadings toward a Gaussian distribution following the central limit theorem. In addition, by ascribing equal significance to neurons with small loadings and neurons with large loadings, susceptibility to noise increases, which is typically Gaussian distributed (*4*). In contrast, SCA addresses these shortcomings, making it well suited for detecting sparse implementation structures and investigating datasets with rich temporal modulation and higher-dimensional stimulus coding.

By exploiting neuronal implementation, SCA identifies a unique set of activity components without assuming the underlying activity patterns (instead assuming sparse contributions). SCA can therefore capture a more complete set of activity variables (dimensions), most notably the temporal modulation of stimulus coding. This reduces bias otherwise introduced by selecting specific time windows, across which neuronal activity is averaged, and

acknowledges the role of different response dynamics for information coding (*20*, *35*). Furthermore, incorporating temporal modulation renders analyses more robust to noise (*36*), which is usually Gaussian and could hide the structure in implementation.

The implementation′s sparse structure is a result of biological constraints regarding the connections among individual neurons. The approximately 10^4 dendritic spines on each cortical neuron (*15*) define an upper limit for the number of neurons it could read out from. The 10^9 neurons in a cortical region such as human PFC (*13*, *14*) and even submodules with one to two magnitudes fewer neurons therefore cannot be reached directly.

Sparse implementations are more efficient in terms of the required energetic demand for establishing readout connections. Neurons are not randomly connected. It is reasonable that the readout neurons would connect to the strongest representing neurons, especially when the animals have been extensively trained on a task and neural plasticity allows for adopting a more efficient code. A dense representation on the other hand would entail wasting considerable amounts of computational resources on the neurons with no appropriate connections to the readout neurons. This would prevent representing a wide range of other information in the same neuronal population.

The addition of one connection step would allow reaching the majority of PFC neurons but at the cost of producing a layer of 10^4 to 10^5 neurons that are dedicated exclusively to feeding the single hypothetical downstream neuron. This is prohibitively inefficient. In such polysynaptic chains, it is more likely that meaningful representations have already emerged in intermediate layers as a result of direct connections from the source region. This notion is also in line with the high dimensionality and nonlinear mixed selectivity characteristic of PFC, which allow for direct linear readout of complex representations without further computations (*7*).

Neurons share inputs and have local recurrent connections, which are particularly pronounced in association cortices such as the PFC (*32*), resulting in more similar firing patterns among neurons within cortical regions. Consequently, neurons might display activity that is weakly correlated to some components of the representational geometry, although they do not participate in the readout. This emphasizes the importance of truncating neurons with weak loadings and enforcing sparsity constraints for estimating potential readout connections (Fig. 4) and motivates the use of dynamical systems modeling to validate correlative measures (Fig. 7).

Beyond reducing energetic demands, sparse implementations offer additional advantages. First, sparsity minimizes the neuronal overlap between representations of distinct stimuli, thereby mitigating interference by segregating inputs across distinct neuronal groups. Second, sparsity calibrates the trade-off between discrimination and generalization, i.e., allowing neural systems to differentiate similar inputs while maintaining consistent responses to noisy variations of the same input (*37*). We identified approximately 30 dominant neurons for each component out of nearly 500 recorded neurons, a level of sparsity that aligns well with previously suggested optimal levels necessary to balance discrimination and generalization for efficient cognitive processing (*37*).

Last, the degree of sparsity might reflect an animal′s training experience and behavioral strategy. Previous studies have found that in the inferotemporal lobe, familiar stimuli are more sparsely represented than novel stimuli (*38*, *39*). Conversely, in the PFC, flexible behaviors are associated with sparser representations of task information than behaviors that are repeated routinely (*3*). These findings suggest that the neuronal implementation is indicative of the specific computations unique to each cortical region.

Working memory persistence without neuronal persistence

Applied to working memory maintenance in the face of distraction, our framework uncovered a sequential representation of numerosity information across multiple task epochs (Fig. 2). This result was neither encouraged nor guaranteed by SCA. This suggests that the readout of memory content from the PFC is optimized for accuracy in each behavioral context rather than optimized for stability across task epochs. The distractor occupied the same resources as the sample numerosity with regard to the sensory and memory component (Fig. 3). Subsequently, an additional component, the recovery component, was recruited to maintain sample information and potentially provide a more response-potent representation (*12*, *40*). Thus, working memory content was maintained by distinct mechanisms before and after interference (Fig. 4).

The subpopulation of recovery neurons was characterized by spiking properties that set these neurons apart from the other populations and could render them particularly suited to working memory storage. Their longer intrinsic time scales (Fig. 5) suggest more stable memory retention (*26*, *29*, *41*) and are consistent with their representing memory content later in the trial (*26*). These neurons also distinguished between sample and distractor contexts, which is crucial for determining what information to keep and what information to discard (Fig. 6). The contextual signal was additively mixed with the numerosity coding signal in these neurons but might still act as gain modulation for numerosity information given the neuronal input-output nonlinearity (*42*).

Representing memory content by sequentially active subpopulations is advantageous. With relay of information, a result of locally feed-forward connectivity, a network can maintain multiple inputs from previous time points and show more resistance to noise (*43*). Furthermore, the PFC might be nonlinearly mixing context and memory representations in all possible ways, expanding dimensionality to enable flexible readout (*7*). Extensive training could have strengthened the nonlinear mixture of second memory epoch context and sample numerosity representations that was most important in the current task, with the PFC retaining other mixtures (e.g., the component coding for sample numerosity in the first memory epoch) for other behavioral demands. In this view, the subpopulation of memory neurons could function as a more passive short-term memory storage oblivious to the behavioral relevance of the memorized information.

We note that our finding of sequentially active components does not argue against stable working memory representations (*24*, *44*, *45*). The memory and recovery components stably encoded working memory content within task epochs (Fig. 4). Working memory representations across epochs, however, were not implemented by a single subpopulation of neurons but three distinct subpopulations. Introducing distraction into the memory delay unmasked the crucial role of recovery neurons for working memory maintenance, which would have been hidden in simpler tasks without distractors. Unlike in previous studies (*11*, *12*), our numerical stimuli were cognitively more demanding, the distractor was presented in the same visual format as the sample, and the distractor was not explicitly cued. These could have resulted in a stronger effect of interference that fully occupied the memory

component and forced sequential representations, whereas in other task designs, the level of distraction might still have allowed neurons to continuously represent sample information across the trial (*11*, *12*). This highlights the importance of including richer temporal structure, multiple processing stages, and behavioral perturbation into cognitive task designs to enable dissection of higher-order brain functions in finer detail and sampling from the full spectrum of underlying mechanisms.

Alternative implementation structures

We focused here on detecting sparse structure in the representational geometry's neuronal implementation, which is linked to the standardized moment of kurtosis. Consequently, the loading distributions have both positive and negative heavy tails. Reading out a given SC thus requires both excitatory and inhibitory connections. However, long-range corticocortical projections are mainly excitatory. This means that other selection criteria that capture nonsymmetrical structure such as the standardized moment of skewness should also be explored (*46*, *47*).

Structure could be in the form of disjointed cell clusters (*34*) or a mixture of Gaussians (*42*). However, if present, these structures would not dissect the representational geometry, as they do not have a one-to-one relation to the dimensions in the activity space. Our neuronal implementation followed a unimodal Laplace distribution (Fig. 2H) instead of a multimodal distribution.

Structure can also be investigated when there are no prior assumptions about the underlying distributions of neuronal loadings. For example, given that neuronal firing is energy-consuming and non-negative, possibly encouraging neurons to align to the dimensions of the representational geometry that have shorter ranges of variation, nonuniform distributions of the number of selective neurons across different dimensions can arise (*48*). However, because all neurons are counted equally, structure probed nonparametrically could potentially be clouded by the large number of weakly coding (nondominant) neurons and thus difficult to detect, particularly in PFC (*4*).

Relation of SCA to other linear dimensionality reduction methods

Different linear dimensionality reduction methods based on L2 reconstruction loss will yield comparable representational geometries, but they will not find the same projections of the representational geometry, i.e., the same components or the same coordinate system in which the data are expressed. The PCs of PCA are conveniently orthogonal and ranked by variance (*49*), but usually, neither correspond to task-related components nor align to the activity of individual neurons (*50*). Sparse PCA requires the factors to be linear projections of the original data and thus only utilizes the covariance among neurons. Therefore, it can only capture the components within the linear span of the data (*51*). Truncating the smaller PCs provides denoised signal as a preprocessing step for independent component analysis (ICA) that can infer the independent sources in the signal space (*52*). Its most common form, fastICA, enforces sparsity constraints on the activity of the components, reflecting an assumption about the activity (*53*). In contrast, in SCA, the sparsity constraint is on the neuronal implementation, i.e., the potential readout weights corresponding to the mixing matrix in ICA, reflecting an assumption about the connectivity.

In summary, our study provides a biologically inspired framework to link representational geometries to single neurons. Neuronal representations must be communicated. Information that cannot be accessed by other neurons does not exist. To understand complex neural systems such as the PFC where we lack clear priors about the signal sources, it is paramount to exploit the circuit and wiring motifs that underlie the observed activity patterns.

MATERIALS AND METHODS

Subjects

Two adult male rhesus monkeys (*Macaca mulatta*, 12 and 13 years old) were used for this study. All experimental procedures were in accordance with the guidelines for animal experimentation approved by the national authority, the Regierungspräsidium Tübingen. A detailed description is provided elsewhere (*9*, *10*). Monkeys were implanted with two right-hemispheric recording chambers centered over the principal sulcus of the lateral PFC and the ventral intraparietal area in the fundus of the intraparietal sulcus. This study reports on the PFC data.

Task and stimuli

The animals grabbed a bar to initiate a trial and maintained eye fixation (ISCAN, Woburn, MA) within 1.75° of visual angle of a central white dot. Stimuli were presented on a centrally placed gray circular background subtending 5.4° of visual angle. Following a 500-ms presample (pure fixation) period, a 500-ms sample stimulus containing one to four dots was shown. The monkeys had to memorize the sample numerosity for 2500 ms and compare it to the number of dots (one to four) presented in a 1000-ms test stimulus. Test stimuli were marked by a red ring surrounding the background circle. If the numerosities matched (50% of trials), then the animals released the bar (correct match trial). If the numerosities were different (50% of trials), then the animals continued to hold the bar until the matching number was presented in the subsequent image (correct nonmatch trial). Match and nonmatch trials were pseudorandomly intermixed. Correct trials were rewarded with a drop of water. In 80% of trials, a 500-ms interfering numerosity of equal numerical range was presented between the sample and test stimulus. The interfering numerosity was independent from either the sample or test numerosity and therefore not useful for solving the task. In 20% of trials, a 500-ms gray background circle without dots was presented instead of an interfering stimulus, i.e., trial length remained constant (control condition, blank). Trials with and without interfering numerosities were pseudorandomly intermixed. Stimulus presentation was balanced: A given sample was followed by all interfering numerosities with equal frequency and vice versa. Throughout the monkeys' training on the distractor task, there was never a condition where a stimulus appearing at the time of the distractor was task relevant.

Low-level, non-numerical visual features could not systematically influence task performance (*10*, *28*): In half of the trials, dot diameters were selected at random. In the other half, dot density and total occupied area were equated across stimuli. CORTEX software (National Institute of Mental Health, Bethesda, MD) was used for experimental control and behavioral data acquisition. New stimuli were generated before each recording session to ensure that the animals did not memorize stimulus sequences.

Electrophysiology

Up to eight 1-megohm glass-insulated tungsten electrodes (Alpha Omega, Israel) per chamber and session were acutely inserted through an intact dura with 1-mm spacing. Single units were recorded at random; no attempt was made to preselect for particular response properties (*10*). Signal amplification, filtering, and digitalization were accomplished with the MAP system (Plexon, Dallas, TX). Waveform separation was performed offline (Plexon Offline Sorter).

Data analysis tools

Data analysis was performed with Python using custom scripts based on packages NumPy, SciPy, scikit-learn, TensorFlow2, PyTorch, Matplotlib, and Plotly.

Preprocessing

Single units were included in the analysis if they were recorded in at least four correct trials of each task condition (meaning each unique sample and distractor numerosity combination). This resulted in 467 neurons across 78 sessions recorded in the PFC. Trials without distractors were not included in the analyses unless specified otherwise.

Unless specified otherwise, the firing rates were binned in a Gaussian window with sigma of 50 ms and step of 100 ms, aligned to the start of the fixation period. The data were then organized into a condition–by–time point–by–neuron tensor. Each tensor entry was normalized by the SD across trials (within each condition). This operation was done to better reflect the information represented in the neuronal population, which can be related to between-class covariance over within-class covariance. In our data, the neurons were not simultaneously recorded. We therefore reduced the within-class covariance to each neuron's within-class variance.

Demixing

Given the independence of the task variables sample numerosity (s), distractor numerosity (d), and trial time (t), the neuronal activity can be directly factorized into parts for each variable and their interaction

$$x = \overline{x} + \overline{x_t} + \overline{x_s} + \overline{x}_d + \overline{x}_{st} + \overline{x}_{dt} + \overline{x}_{sd} + \overline{x}_{sdt} \quad (1)$$

where $\overline{x}$ is the mean activity of a neuron (effectively mean-centering the demixed parts).

Because the stimulus response is also modulated by time, each part was grouped together with its interaction with time (*22*)

$$x_{\text{time}} = \overline{x_t} \quad (2)$$

with the dimensionality of 45 (time points; 4.5-s trial length; 100-ms bin width)

$$x_{\text{sample}} = \overline{x_s} + \overline{x}_{st} \quad (3)$$

with the dimensionality of 4 × 45 = 180 (sample numerosities × time points)

$$x_{\text{distractor}} = \overline{x}_d + \overline{x}_{dt} \quad (4)$$

with the dimensionality of 4 × 45 = 180 (distractor numerosities × time points)

$$x_{\text{sd interaction}} = \overline{x}_{sd} + \overline{x}_{sdt} \quad (5)$$

with the dimensionality of 4 × 4 × 45 = 720 (sample numerosities × distractor numerosities × time points).

Visualization of representation and implementation space

For a data matrix X where each column vector x is the demixed activity of a neuron, the singular value decomposition was taken

$$X = U\Sigma V^T \quad (6)$$

where U and V are unitary matrices and Σ is a diagonal matrix with ordered singular values. The first n columns of $U\Sigma$ are the PCs that were used to visualize the representational geometry. The first n columns of $V\Sigma$ are loadings on the PCs that were used to visualize the implementation space.

Within this subspace, an arbitrary component can be specified with $U\Sigma P_{:,1}$ ($P_{:,1}$ being a column vector from a unitary matrix P), with the orientation of this component given by $P_{:,1}$. The loadings on this component will be the first row of $(U\Sigma P)^+X = P^TV^T$, that is ${P_{:,1}}^T V^T$. This way, the loadings are visualized with the same orientation $P_{:,1}$. in implementation space as their corresponding component in representation space.

Sparsity measures

The SI of neuronal loadings $\mathbf{x}$ is given by

$$\text{SI}(\mathbf{x}) = \frac{\text{kurtosis}(\mathbf{x})}{3} \quad (7)$$

$$\text{kurtosis}(\mathbf{x}) = \langle(\mathbf{x} - \overline{\mathbf{x}})^4\rangle / \langle(\mathbf{x} - \overline{\mathbf{x}})^2\rangle^2 \quad (8)$$

SI is thus proportional to kurtosis. High SI reflects heavy tails. Low SI approximates a box-car distribution. A Gaussian distribution's SI is 1.

The sparsity parameter β is calculated by fitting generalized normal distributions to the neuronal loadings. The probability density function of a generalized normal distribution is defined as

$$f_X(x;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} e^{-\left(\frac{|x-\mu|}{\alpha}\right)^\beta} \quad x \in \mathbb{R}, \alpha > 0, \beta > 0 \quad (9)$$

$$\text{where } \Gamma(z) = \int_0^\infty t^{z-1}e^t dt \; z > 0 \quad (10)$$

α controls the variance of the distribution. β controls the kurtosis of the distribution (sparsity). β = 2 yields a Gaussian distribution. The parameters were fitted using maximum likelihood estimation.

Sparse component analysis

Following the formulation of sparse coding (*16*, *17*, *54*), SCA reduces the dimensionality of the dataset and identifies the unique components by enforcing a sparse penalty on neuronal

loadings

$$\text{Loss} = \|X - \sum_{i=1}^{k} \overrightarrow{\mathbf{u}_i}\ \overrightarrow{\mathbf{v}_i}^T\|_{\text{Frobenius}} + \alpha \sum_{i=1}^{k} \|\overrightarrow{\mathbf{v}_i}\|_1 + \beta \sum_{i=1}^{k} \|\overrightarrow{\mathbf{v}_i}\|_2^2 \quad (11)$$

where $\|\overrightarrow{\mathbf{r}_i}\| = 1$.

The loss function is defined as the sum of the reconstruction loss and the regularizations. Data X is organized as an n activity instances (combination of conditions and time points) by p neurons matrix. X is then approximated by k firing activity vectors $\overrightarrow{\mathbf{u}}$ and their corresponding neuronal loadings $\overrightarrow{\mathbf{v}}$. The parameter α controls the strength of L1 regularization that encourages sparsity of the loadings. Parameters α and k were determined by a twofold cross-validated grid search. The L2 regularization coefficient β was set at 0.01 to smooth the loss landscape and make the result stable across random initializations.

Substitute data for SCA

Substitute data were created for the demixed sample coding part X of the data (Fig. 2). For the singular value decomposition $X = U\Sigma V^T$, $U\Sigma$ specifies the representational geometry (see above). Operations were performed on V only.

A random unitary matrix R with the size of the number of neurons was drawn from a Haar distribution. The original matrix V was replaced with $V' = VR$. V' is also a unitary matrix, meaning that this manipulation will not change the geometries but will rotate them to random axes. In other words, it will linearly combine the loadings including those on the components with very low variance, which will render the substitute distribution of loadings on the sample numerosity components close to Gaussian. The substitute data are then

$$X' = U\Sigma V'^T = XR \quad (12)$$

Synthetic data with continuously active SCs

We constructed continuously active components from the sample numerosity representing subspace defined by the activity patterns of three sample numerosity representing SCs (each active in one task epoch; shown in Fig. 2F) denoted as $U_{\text{sequential}}$, where each column of $U_{\text{sequential}}$ is one SC's activity pattern. We can then write arbitrary components in this subspace as $U_{\text{sequential}}Q$, where each column of Q denotes the coefficients of the linear combination for that component. We designed a transformation matrix $Q = [[-1, 1, 1]^T, [-1, 1, 1]^T, [-1, 1, 1]^T]$ that is full rank and has no zero elements to ensure that it spans the same subspace and that each constructed component is active in all task epochs (thus continuous). This transformation matrix was then orthonormalized using the Gram-Schmidt process and yielded $Q' = [[-0.58, 0.58, 0.58]^T, [0.78, 0.21, 0.57]^T, [-0.21, -0.78, 0.57]^T]$. The constructed continuously active components were then

$$U_{\text{continuous}} = U_{\text{sequential}}Q' \quad (13)$$

We then created synthetic neurons using these continuously active components. The neuronal loading on each continuously active component was independently sampled from a generalized normal distribution (see Eq. 9) with parameters matching that of the original data (sparsity β = 1.1, α = 1; the whole distribution was then scaled to match the variance of the SCs in the original data). The synthetic neurons' loadings are denoted as V_{sparse}, where each column vector of V_{sparse} is the neuronal population's loadings on one component (with a length of 467, matching the number of neurons in the original data). The synthetic data are then given by

$$X_{\text{synthetic}} = U_{\text{continuous}} V_{\text{sparse}}^T \quad (14)$$

Measures of SC activity

$\overrightarrow{\mathbf{u}_i}$ in SCA specifies the activity of the SC i. The following measures of the set of $\overrightarrow{\mathbf{u}_i}$ were compared between the original dataset and its substitutes (n = 1000).

Spread of representation

The SD of $\overrightarrow{\mathbf{u}_i}$ across different numerosity conditions k at each time point was used to define the relative (normalized) information at that time point. Specifically, each $\overrightarrow{\mathbf{u}_i}$ was first reshaped into a condition–by–time point matrix Y^i. Then, the information in component i at time point t is given by

$$Z_{i,t} = \sqrt{\langle (Y^i_{k,t} - \langle Y^i_{k,t} \rangle_k)^2 \rangle_k} \quad (15)$$

The skewness of the information across time points was calculated for each component and averaged across components as follows

$$\text{Skew}_i = \langle (Z_{i,t} - \overline{Z_{i,t}})^3 \rangle_t / \langle (Z_{i,t} - \overline{Z_{i,t}})^2 \rangle_t^{3/2} \quad (16)$$

Positively skewed Z indicates a long tail in the distribution of information across time points, corresponding to a few time points having high information. Conversely, a smaller or even negative skewness implies that there are more high-information time points than low-information time points, making the high information more spread out across time points. We define the spread of representation as the negative skewness

$$\text{Spread} = -\langle \text{Skew}_i \rangle_i \quad (17)$$

Overlap of active periods

The dot product of the information of every pair of components i and j was taken and averaged across pairs

$$\text{Overlap} = \langle Z_{i,t} Z^T_{j,t} \rangle \quad (18)$$

Maximum tuning reversal

A given component i may show changes of tuning to sample numerosities during the course of a trial. Its tuning at time t is specified by $Y^i_{:,t}$. For each component i, the dot product similarity of tunings between time point pairs was specified in the nondiagonal entries in $C^i = Y^{iT}Y^i$, where the diagonal entries are the strength of the tuning at each time point. C^i was then normalized to the strongest tuning: $C^{i\prime} = C^i/\max(C^i)$. The most negative entry in $C^{i\prime}$ was then the degree of reversal in this component. $\text{Reversal}_i = -\min(C^{i\prime})$. It would reach the maximum of 1 when tuning at a given time point is the complete reversal of the strongest tuning. It would be close to 0 when the tuning does not reverse. The maximum tuning reversal is

then the largest reversal in a set of SCs

$$\text{Max tuning reversal} = \max_i \text{Reversal}_i = \max_i \left\{ -\min \left[\frac{Y^{i^T} Y^i}{\max(Y^{i^T} Y^i)} \right] \right\} \quad (19)$$

Component similarity

Let U_{sca} be the concatenation of activity $\vec{\mathbf{u}}_i$ and V_{sca} the concatenation of loadings $\vec{\mathbf{v}}_i$ of the SC i. The data matrix can be expressed as $X = U_{sca} V_{sca}^T + \epsilon$. ϵ denotes the noise term. Then, it follows $U_{sca}^+(X - \epsilon) = V_{sca}^T$. The pseudoinverse U_{sca}^+ can be viewed as a linear transform of the original data. Since all the activities $\vec{\mathbf{u}}$ have unit length, larger loadings would be required to express an arbitrary geometry when the activities are correlated, meaning lower efficiency. The component similarity is measured by the product of the singular values of U_{sca}. Formally, if the singular value decomposition gives $U_{sca} = U\Sigma V^T$, then

$$\text{Similarity} = \prod_i \Sigma_{i,i} \quad (20)$$

The similarity can also be viewed as the determinant of the transformation matrix from arbitrary orthogonal bases to the bases of U_{sca}.

Numerosity information in different components

The SD $Z_{i,t}$ for all time points t specifies the evolution of normalized information within this component, but because $\vec{u}_i$ in component i has unit length, this measure does not allow for direct comparisons between components (see above). To allow for such comparisons (fig. S2), the norm of $\vec{v_i}$ is therefore applied to $Z_{i,t}$ as a scaling factor

$$\text{Information} = \|\vec{v_i}\| Z_{i,t} \quad (21)$$

LDA decoding

Neurons recorded in different sessions were stitched together. To account for the different number of trials recorded per neuron, a criterion was set to ensure that there were at least 1.5 times more trials than neurons. This resulted in 228 neurons with at least 385 trials each. Removing incorrect trials and selecting the minimum number of trials recorded per condition and neuron left 118 trials per neuron. Trials of the same condition were then randomly selected for each repetition of the analysis.

Multiclass LDA (scikit-learn package) was used for decoding because of its advantageous property of accounting for data covariance. LDA assumes the same covariance in every class. It finds the projection that preserves the Mahalanobis distance between classes and predicts the label of a new data point by its Mahalanobis distance to the class centroid. Shrinkage of the measured covariance matrix was performed by averaging with a diagonal matrix. The strength of shrinkage was determined following the Ledoit-Wolf lemma (*55*). Decoding accuracy, i.e., the ratio of correctly predicted trials, was averaged across seven repetitions of sevenfold cross-validation.

Spike train statistics

Firing rates were binned in a Gaussian window with sigma of 12.5 ms and step of 25 ms.

Correlation, autocorrelation, and intrinsic time scales were determined as described elsewhere (*29*). The firing rate of each neuron n at time point t of trial i is expressed as $x_{n,i,t}$. The Pearson correlation between time points $t1$ and $t2$ is then

$$r_n(t1, t2) = \frac{\langle (x_{n,i,t1} - \langle x_{n,i,t1} \rangle_i)(x_{n,i,t2} - \langle x_{n,i,t2} \rangle_i) \rangle_i}{\langle (x_{n,i,t1} - \langle x_{n,i,t1} \rangle_i)^2 \rangle_i^{1/2} \langle (x_{n,i,t2} - \langle x_{n,i,t2} \rangle_i) \rangle_i^{1/2}} \quad (22)$$

Autocorrelation is defined as

$$AC_n(\Delta t) = \langle r_n(t0, t0 + \Delta t) \rangle_{t0} \quad (23)$$

To account for the refractoriness and adaptation at small time lags, fitting started at the time lag where the autocorrelation function had dropped most strongly. Neurons with the strongest drop after 400 ms were discarded (six neurons). The autocorrelation was then fitted with an exponential decay

$$AC(\Delta t) = A[\exp(-\Delta t/\tau) + B] \quad (24)$$

Parameters A and B were constrained in [0,1], and τ was constrained from 10 to 2000 ms. The autocorrelation function of eight neurons could not be fitted. The neurons with τ fitted below 20 ms (20 neurons) or above 1600 ms (25 neurons) were excluded because of the biologically unrealistic fit. This left 408 neurons. Very few neurons were excluded in the dominant subpopulations (two neurons, two neurons, and one neuron for the sensory, memory, and recovery subpopulation, respectively).

The ISIs were determined for the entire session. The CV measures the global variation of a neuron's ISI and is defined as

$$\text{CV} = \text{s.d.}(\text{ISI}) / \langle \text{ISI} \rangle \quad (25)$$

In contrast to CV, LV measures the local ISI change (*56*). It is defined as

$$\text{LV} = \frac{3}{n-1} \sum_{i=1}^{n-1} (\text{ISI}_i - \text{ISI}_{i+1})^2 / (\text{ISI}_i + \text{ISI}_{i+1})^2 \quad (26)$$

CV and LV are both expected to be 1 for spiking activity following a Poisson process. CV and LV would be 0 for perfectly regular firing and larger than 1 for more irregular firing than by a Poisson process.

Kullback-Leibler divergence

KL divergence measures the difference between two distributions. For the analyses of intrinsic time scales and periodicity, KL divergence was calculated between the distribution of statistic x for the entire population P and that of subsamples Q (either dominant subpopulations or bootstrap subsamples). It is given by

$$D_{KL}(P \| Q) = -\sum\nolimits_x P(x) \cdot \log Q(x)/P(x) \quad (27)$$

To create the null distribution of D_{KL}, 27 neurons (comparable to the number of neurons in the dominant subpopulations after exclusion of neurons in which no autocorrelation function could be fitted) were randomly sampled from the PFC population 1000 times.

Temporal dynamics

Periodicity

The Fourier transform of the demixed temporal part of the firing rate of each neuron is given by

$$\mathrm{PSD}(f) = \mathrm{DFT}[x_{\mathrm{time}}(t)] \tag{28}$$

Then, the periodicity was defined as the ratio between the power of the harmonics of 1/1.5 Hz (reflecting the onset of visual input at regular spacing of 1.5 s) and the power of all frequencies

$$\mathrm{Periodicity} = \sum\nolimits_{i\in Z^+} \mathrm{PSD}\left(i\frac{2}{3}\right) / \sum\nolimits_f \mathrm{PSD}(f) \tag{29}$$

Tangling

Tangling reflects the smoothness and stability of the flow field around the vicinity of state x_t on a trajectory (*30*). It is given by

$$Q(t) = \max_{t'} \frac{\left\|\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_{t'}\right\|^2}{\left\|\mathbf{x}_t - \mathbf{x}_{t'}\right\|^2 + \epsilon} \tag{30}$$

It specifies the maximum difference between the derivative at state x_t and the derivative at other states $\mathbf{x}_{t'}$, normalized by their Euclidean distance. A small constant ϵ was added to avoid numerical error when the two states were too close.

Recurrent neural network

An RNN model was implemented using the PyTorch neural network module. The model has the formulation

$$\mathbf{r}(s,d,t+1) = \phi[W\mathbf{r}(s,d,t) + I\mathbf{n}(s,d,t) + \mathbf{b}] \tag{31}$$

$\boldsymbol{r}$ is the firing rate of units in the condition of sample numerosity s and distractor numerosity d at time point t. ϕ is the nonlinear activation function, chosen to be a rectified linear unit to respect the biological characteristics of non-negative firing rates with high upper limits. W is the within-population connectivity matrix. I is the input matrix with the dimensions of 467 (total number of units) by 4 (number of numerosities). A column $I_{:,a}$ is the input to the units when numerosity a is being presented. $\mathbf{n}$ is an indicator vector with the entry $\mathbf{n}_a$ corresponding to the presented numerosity being 1 and all other entries being 0. $\mathbf{b}$ is the intercept. W, I, and $\mathbf{b}$ are the parameters to be trained. Formally, $\mathbf{n}$ as a function of trial type specified by s and d and time point t is defined by

$$\mathbf{n}(s,d,t) = \mathbf{m}(s)\cdot \mathrm{mask}_{[0.5,1)}(t) + \mathbf{m}(d)\cdot \mathrm{mask}_{[2,2.5)}(t) \tag{32}$$

$$\text{where } \mathbf{m}(x) = [\mathbf{1}_{\{1\}}(x), \mathbf{1}_{\{2\}}(x), \mathbf{1}_{\{3\}}(x), \mathbf{1}_{\{4\}}(x)]^T \tag{33}$$

$$\mathrm{mask}_A(t) = \mathbf{1}_A(t*0.1) \tag{34}$$

$$\mathbf{1}_A(x) := \begin{cases} 1, x \in A \\ 0, x \notin A \end{cases} \tag{35}$$

$\mathbf{m}$ maps a numerosity to the corresponding one-hot vector. $\mathrm{mask}_A(t)$ indicates the time (0.1 s steps) when the corresponding stimulus is presented. $\mathbf{1}_A(x)$ is an ancillary indicator function to define $\mathbf{m}$ and mask.

The model was trained to produce the whole sequence of firing rates $\mathbf{r}(s, d, t)$ to match the target data $\mathbf{x}_{s,d,t}$, given the initial firing rate in the fixation period $\mathbf{r}(s, d, t_0)$ and the input $\mathbf{n}(s, d, t)$. The loss function is defined as

$$\mathrm{Loss}(W,I,\mathbf{b}) = \sum_{s,d,t} [\mathbf{r}(s,d,t) - \mathbf{x}_{s,d,t}]^2 + \lambda\|W\|_1 + \lambda\|I\|_1 \tag{36}$$

$$\mathbf{r}(s,d,t_0) = \mathbf{x}_{s,d,t_0} \tag{37}$$

The coefficient λ controls the strength of regularization and was determined by a grid search with cross validation. The model weights were initialized by sampling from the uniform distribution $[-\sqrt{N}, \sqrt{N}]$, where N is the number of neurons. Because the initial model weights influence the learning result (*57*), we chose the densest distribution to not bias the result toward sparse implementations.

The prediction of the later time points relies on the quality of the prediction of the early time points. If the training was done only by giving the first time point, then convergence would be difficult to achieve and learning would be heavily biased toward reproducing early time points in the data. To overcome this possible instability, the model was trained in a recursive fashion by first using every time point as the initial firing rate, training the model to predict the following time points, and gradually increasing the number of time points the model needs to predict. Hence, at each iteration i, the temporal sequence $x_{s,d,t}$ was reorganized into $T - i$ chunks of length $i + 1$, $\{\mathbf{x}_{s,d,t_0}, \ldots, \mathbf{x}_{s,d,t_0+i}\}$ where $t_0 \in \{1, \ldots, T - i\}$, with the first firing rate in each chunk as initial firing rate and the rest as target to be fit by the model.

Variance explained by RNN

The variance explained by the model was determined by the difference between the model's predicted trajectory and the trajectory of the original data normalized to the difference between a reference trajectory (constant activity set to the first entry of the fixation period) and the trajectory of the original data

$$\mathrm{EV} = 1 - \sum_{s,d,t} [\mathbf{r}(s,d,t) - \mathbf{x}_{s,d,t}]^2 / \sum_{s,d,t} [\mathbf{x}_{s,d,t_0} - \mathbf{x}_{s,d,t}]^2 \tag{38}$$

The normalized EV (Fig. 7C, right axis) was defined as the difference between a substitute's EV and the original data's EV, divided by the percentage of the manipulated variance (numerosity coding signal, 27.4%; compare to Fig. 2B). EV for the numerosity signal (Fig. 7D) was calculated by replacing both $\mathbf{r}(s, d, t)$ and $\mathbf{x}_{s,d,t}$ with their demixed numerosity representing parts.

Substitute data for RNN

To not distort the strong connection between sample and distractor numerosity coding (e.g., Fig. 3B and figs. S2 and fig. S5B), the loadings of these two parts of the data and their interaction were shuffled together to create three types of substitute datasets. The RNN model was then trained on the substitutes.

Gaussian distribution of loadings

The Gaussian substitutes were created as described for SCA, except for that singular value decomposition that was performed on $X_{\mathrm{sample}} + X_{\mathrm{distractor}} + X_{\mathrm{sd\text{-}interaction}} = X_{\mathrm{all}} - X_t = U\Sigma V^T$.

Sparse distribution with random alignment

For k dimensions of the numerosity coding part of the data (determined by cross-validation), a $k \times k$ unitary matrix R was randomly

drawn from a Haar distribution and combined with an identity matrix I to create $R' = \begin{pmatrix} R & 0 \\ 0 & I \end{pmatrix}$. Then, $V' = VR'$ was substituted for V. This leaves the sparse structure in the original k dimensional numerosity representing subspace intact but rotates the sparse structure in $V_{:,1:k}$ to random orientations.

Sparse distribution with original alignment

The rows of $V_{:,1:k}$, i.e., the neuronal identities, were permuted by substituting $V' = (V_{\text{permute},1:k}, V_{:,k+1:p})$ for V.

Supplementary Materials

This PDF file includes:

Figs. S1 to S5

REFERENCES AND NOTES

1. D. L. Barack, J. W. Krakauer, Two views on the cognitive brain. *Nat. Rev. Neurosci.* **22**, 359–371 (2021).
2. S. Saxena, J. P. Cunningham, Towards the neural population doctrine. *Curr. Opin. Neurobiol.* **55**, 103–111 (2019).
3. F.-K. Chiang, J. D. Wallis, E. L. Rich, Cognitive strategies shift information from single neurons to populations in prefrontal cortex. *Neuron* **110**, 709–721.e4 (2022).
4. S. Bernardi, M. K. Benna, M. Rigotti, J. Munuera, S. Fusi, C. D. Salzman, The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell* **183**, 954–967.e21 (2020).
5. G. Okazawa, C. E. Hatch, A. Mancoo, C. K. Machens, R. Kiani, Representational geometry of perceptual decisions in the monkey parietal cortex. *Cell* **184**, 3748–3761.e18 (2021).
6. N. Kriegeskorte, X.-X. Wei, Neural tuning and representational geometry. *Nat. Rev. Neurosci.* **22**, 703–718 (2021).
7. M. Rigotti, O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, S. Fusi, The importance of mixed selectivity in complex cognitive tasks. *Nature* **497**, 585–590 (2013).
8. S. E. Cavanagh, J. P. Towers, J. D. Wallis, L. T. Hunt, S. W. Kennerley, Reconciling persistent and dynamic hypotheses of working memory coding in prefrontal cortex. *Nat. Commun.* **9**, 3498 (2018).
9. S. N. Jacob, D. Hähnke, A. Nieder, Structuring of abstract working memory content by fronto-parietal synchrony in primate cortex. *Neuron* **99**, 588–597.e5 (2018).
10. S. N. Jacob, A. Nieder, Complementary roles for primate frontal and parietal cortex in guarding working memory from distractor stimuli. *Neuron* **83**, 226–237 (2014).
11. A. Parthasarathy, C. Tang, R. Herikstad, L. F. Cheong, S.-C. Yen, C. Libedinsky, Time-invariant working memory representations in the presence of code-morphing in the lateral prefrontal cortex. *Nat. Commun.* **10**, 4995 (2019).
12. C. Tang, R. Herikstad, A. Parthasarathy, C. Libedinsky, S.-C. Yen, Minimally dependent activity subspaces for working memory and motor preparation in the lateral prefrontal cortex. *eLife* **9**, e58154 (2020).
13. E. Courchesne, P. R. Mouton, M. E. Calhoun, K. Semendeferi, C. Ahrens-Barbeau, M. J. Hallet, C. C. Barnes, K. Pierce, Neuron number and size in prefrontal cortex of children with autism. *JAMA* **306**, 2001–2010 (2011).
14. S. Herculano-Houzel, K. Catania, P. R. Manger, J. H. Kaas, Mammalian brains are made of these: A dataset of the numbers and densities of neuronal and nonneuronal cells in the brain of glires, primates, scandentia, eulipotyphlans, afrotherians and artiodactyls, and their relationship with body mass. *Brain Behav. Evol.* **86**, 145–163 (2015).
15. G. Eyal, M. B. Verhoog, G. Testa-Silva, Y. Deitcher, R. Benavides-Piccione, J. DeFelipe, C. P. J. de Kock, H. D. Mansvelder, I. Segev, Human cortical pyramidal neurons: From spines to spikes via models. *Front. Cell. Neurosci.* **12**, 181 (2018).
16. P. Georgiev, F. Theis, A. Cichocki, H. Bakardjian, *Sparse Component Analysis: A New Tool for Data Mining* (Data Mining in Biomedicine, Springer, 2007), vol. 7, pp. 91–116.
17. B. A. Olshausen, D. J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996).
18. M. C. Aoi, V. Mante, J. W. Pillow, Prefrontal cortex exhibits multidimensional dynamic encoding during decision-making. *Nat. Neurosci.* **23**, 1410–1420 (2020).
19. A. Libby, T. J. Buschman, Rotational dynamics reduce interference between sensory and memory representations. *Nat. Neurosci.* **24**, 715–726 (2021).
20. V. Mante, D. Sussillo, K. V. Shenoy, W. T. Newsome, Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84 (2013).
21. S. Kornblith, M. Norouzi, H. Lee, G. Hinton, "Similarity of neural network representations revisited." In International conference on machine learning, pp. 3519-3529. PMLR (2019).
22. D. Kobak, W. Brendel, C. Constantinidis, C. E. Feierstein, A. Kepecs, Z. F. Mainen, X.-L. Qi, R. Romo, N. Uchida, C. K. Machens, Demixed principal component analysis of neural population data. *eLife* **5**, e10989 (2016).
23. G. F. Elsayed, J. P. Cunningham, Structure in neural population recordings: An expected byproduct of simpler phenomena? *Nat. Neurosci.* **20**, 1310–1318 (2017).
24. J. D. Murray, A. Bernacchia, N. A. Roy, C. Constantinidis, R. Romo, X.-J. Wang, Stable population coding for working memory coexists with heterogeneous neural dynamics in prefrontal cortex. *Proc. Natl. Acad. Sci. U.S.A.* **114**, 394–399 (2017).
25. A. Parthasarathy, R. Herikstad, J. H. Bong, F. S. Medina, C. Libedinsky, S.-C. Yen, Mixed selectivity morphs population codes in prefrontal cortex. *Nat. Neurosci.* **20**, 1770–1779 (2017).
26. D. F. Wasmuht, E. Spaak, T. J. Buschman, E. K. Miller, M. G. Stokes, Intrinsic neuronal dynamics predict distinct functional roles during working memory. *Nat. Commun.* **9**, 3499 (2018).
27. E. Spaak, K. Watanabe, S. Funahashi, M. G. Stokes, Stable and dynamic coding for working memory in primate prefrontal cortex. *J. Neurosci.* **37**, 6503–6516 (2017).
28. A. Nieder, D. J. Freedman, E. K. Miller, Representation of the quantity of visual items in the primate prefrontal cortex. *Science* **297**, 1708–1711 (2002).
29. J. D. Murray, A. Bernacchia, D. J. Freedman, R. Romo, J. D. Wallis, X. Cai, C. Padoa-Schioppa, T. Pasternak, H. Seo, D. Lee, X.-J. Wang, A hierarchy of intrinsic timescales across primate cortex. *Nat. Neurosci.* **17**, 1661–1663 (2014).
30. A. A. Russo, S. R. Bittner, S. M. Perkins, J. S. Seely, B. M. London, A. H. Lara, A. Miri, N. J. Marshall, A. Kohn, T. M. Jessell, L. F. Abbott, J. P. Cunningham, M. M. Churchland, Motor cortex embeds muscle-like commands in an untangled population response. *Neuron* **97**, 953–966.e8 (2018).
31. S. Ladjal, A. Newson, C.-H. Pham, A PCA-like autoencoder. arXiv:1904.01277 [quant-ph] (2019).
32. J. A. Harris, S. Mihalas, K. E. Hirokawa, J. D. Whitesell, H. Choi, A. Bernard, P. Bohn, S. Caldejon, L. Casal, A. Cho, A. Feiner, D. Feng, N. Gaudreault, C. R. Gerfen, N. Graddis, P. A. Groblewski, A. M. Henry, A. Ho, R. Howard, J. E. Knox, L. Kuan, X. Kuang, J. Lecoq, P. Lesnar, Y. Li, J. Luviano, S. McConoughey, M. T. Mortrud, M. Naeemi, L. Ng, S. W. Oh, B. Ouellette, E. Shen, S. A. Sorensen, W. Wakeman, Q. Wang, Y. Wang, A. Williford, J. W. Phillips, A. R. Jones, C. Koch, H. Zeng, Hierarchical organization of cortical and thalamic connectivity. *Nature* **575**, 195–202 (2019).
33. S. Chung, L. F. Abbott, Neural population geometry: An approach for understanding biological and artificial neural networks. *Curr. Opin. Neurobiol.* **70**, 137–144 (2021).
34. J. Hirokawa, A. Vaughan, P. Masset, T. Ott, A. Kepecs, Frontal cortex neuron types categorically encode single decision variables. *Nature* **576**, 446–451 (2019).
35. G. Bondanelli, S. Ostojic, Coding with transient trajectories in recurrent neural networks. *PLOS Comput. Biol.* **16**, e1007655 (2020).
36. I. M. Johnstone, A. Y. Lu, On consistency and sparsity for principal components analysis in high dimensions. *J. Am. Stat. Assoc.* **104**, 682–693 (2009).
37. O. Barak, M. Rigotti, S. Fusi, The sparseness of mixed selectivity neurons controls the generalization-discrimination trade-off. *J. Neurosci.* **33**, 3844–3856 (2013).
38. L. Woloszyn, D. L. Sheinberg, Effects of long-term visual experience on responses of distinct classes of single units in inferior temporal cortex. *Neuron* **74**, 193–205 (2012).
39. D. J. Freedman, M. Riesenhuber, T. Poggio, E. K. Miller, Experience-dependent sharpening of visual shape selectivity in inferior temporal cortex. *Cereb. Cortex* **16**, 1631–1644 (2006).
40. D. B. Ehrlich, J. D. Murray, Geometry of neural computation unifies working memory and planning. *Proc. Natl. Acad. Sci.* **119**, e2115610119 (2022).
41. R. Kim, T. J. Sejnowski, Strong inhibitory signaling underlies stable temporal dynamics and working memory in spiking neural networks. *Nat. Neurosci.* **24**, 129–139 (2021).
42. A. Dubreuil, A. Valente, M. Beiran, F. Mastrogiuseppe, S. Ostojic, The role of population structure in computations through neural dynamics. *Nat. Neurosci.* **25**, 783–794 (2022).
43. A. E. Orhan, X. Pitkow, "Improved memory in recurrent neural networks with sequential non-normal dynamics." *in 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia*, April 26-30, 2020 (2020); https://openreview.net/forum?id=ryx1wRNFvB.
44. S. Funahashi, C. J. Bruce, P. S. Goldman-Rakic, Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *J. Neurophysiol.* **61**, 331–349 (1989).
45. D. Mendoza-Halliday, S. Torres, J. C. Martinez-Trujillo, Sharp emergence of feature-selective sustained activity along the dorsal visual pathway. *Nat. Neurosci.* **17**, 1255–1262 (2014).
46. V. Koren, A. R. Andrei, M. Hu, V. Dragoi, K. Obermayer, Pairwise synchrony and correlations depend on the structure of the population code in visual cortex. *Cell Rep.* **33**, 108367 (2020).
47. M. Román Rosón, Y. Bauer, A. H. Kotkat, P. Berens, T. Euler, L. Busse, Mouse dLGN receives functional input from a diverse population of retinal ganglion cells with limited convergence. *Neuron* **102**, 462–476.e8 (2019).

48. J. C. Whittington, W. Dorrell, S. Ganguli, T. E. Behrens, Disentangling with biological constraints: A theory of functional cell types. arXiv:2210.01768 [quant-ph] (2022).
49. V. Q. Vu, J. Lei, Minimax sparse principal subspace estimation in high dimensions. *Ann. Statist.* **41**, 2905–2947 (2013).
50. I. Higgins, L. Chang, V. Langston, D. Hassabis, C. Summerfield, D. Tsao, M. Botvinick, Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. *Nat. Commun.* **12**, 6456 (2021).
51. H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis. *J. Comput. Graph. Stat.* **15**, 265–286 (2006).
52. A. Hyvärinen, E. Oja, Independent component analysis: Algorithms and applications. *Neural Netw.* **13**, 411–430 (2000).
53. A. Hyvarinen, *Fast ICA for Noisy Data using Gaussian Moments* (IEEE, 1999), vol. 5, pp. 57–61.
54. H. Lee, A. Battle, R. Raina, A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems* (2007), pp. 801–808.
55. O. Ledoit, M. Wolf, A well-conditioned estimator for large-dimensional covariance matrices. *J. Multivar. Anal.* **88**, 365–411 (2004).
56. S. Shinomoto, H. Kim, T. Shimokawa, N. Matsuno, S. Funahashi, K. Shima, I. Fujita, H. Tamura, T. Doi, K. Kawano, N. Inaba, K. Fukushima, S. Kurkin, K. Kurata, M. Taira, K.-I. Tsutsui, H. Komatsu, T. Ogawa, K. Koida, J. Tanji, K. Toyama, Relating neuronal firing patterns to functional differentiation of cerebral cortex. *PLoS Comput. Biol.* **5**, e1000433 (2009).
57. T. Flesch, K. Juechems, T. Dumbalska, A. Saxe, C. Summerfield, Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron* **110**, 1258–1270.e11 (2022).

Acknowledgments: We thank C. Leibold and M. Grosse-Wentrup for helpful suggestions and comments regarding the implementation framework and data analysis. **Funding:** This work was supported by German Research Foundation (DFG) grants JA 1999/1-1, JA 1999/5-1, and JA 1999/6-1 to S.N.J. and grants NI 618/10-1 and NI 618/13-1 to A.N., as well as European Research Council (ERC H2020) grant GA 758032 to S.N.J. **Author contributions:** Conceptualization: X.X.L. and S.N.J. Methodology: X.X.L. Data collection: S.N.J. and A.N. Data analysis: X.X.L. Data visualization: X.X.L. and S.N.J. Writing (original draft): X.X.L. and S.N.J. Writing (review and editing): X.X.L., A.N., and S.N.J. Supervision: S.N.J. **Competing interests:** The authors declare that they have no competing interest. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials (https://doi.org/10.5061/dryad.j0zpc86m9).

Submitted 20 March 2023
Accepted 14 November 2023
Published 13 December 2023
10.1126/sciadv.adh8685

ScienceAdvances

The neuronal implementation of representational geometry in primate prefrontal cortex

Xiao-Xiong Lin, Andreas Nieder, and Simon N. Jacob

Sci. Adv. **9** (50), eadh8685. DOI: 10.1126/sciadv.adh8685

Science Advances (ISSN 2375-2548) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Advances* is a registered trademark of AAAS.