A parsimonious model for learning order relations provides a principled explanation of diverse experimental data

(March 17, 2025)

Yukun Yang and Wolfgang Maass

Institute of Machine Learning and Neural Computation, Graz University of Technology, Graz, 8010, Austria.

Contributing authors: yukun.yang@tugraz.at; maass@igi.tugraz.at;

Abstract

A cornerstone of higher cognitive function is the ability to learn relations between objects, and to use this relational information for inference. A concrete challenge is learning an order relation, and the use of resulting internal representations for transitive inference. It is known that order relations are represented in the parietal cortex both by a summation code, where the firing rates of neurons indicate the rank of an item in the order, and by a heterogeneous code where neurons are selective for specific rank values. But it remains open how these neural codes are learned. We show that the summation code emerges through a simple rule for synaptic plasticity in a single layer of synaptic connections. The resulting neural representation enables transitive inference and gives rise to the terminal item effect observed in human behavior. We also show that the simultaneous presence of a heterogeneous code gives rise to the commonly observed curvature of 2D projections of fMRI (functional magnetic resonance imaging) signals for items of increasing rank: They tend to lie on a horseshoe-shaped curved line. Our models are supported by a rigorous theory.

1 Introduction

A hallmark of higher cognitive function is the capability to extract and represent information about relations between items from complex sensory inputs, and to use this relational information for inference. In particular, the brain is able to extract information about order relations between items. Furthermore, it encodes it in a way that supports transitive inference, which can be seen a basic prototype for abstract reasoning. Essential for transitive inference is the extraction and neural representation of a latent variable for items in an order relation: their rank. Experimental data show that brains are able to extract this latent variable from dispersed information about order relations between pairs of items, and to encode it in the parietal cortex, in fact in two different ways: By a summation code where many neurons fire with a rate that is indicative of the rank of an item in the order, and by neurons that fire preferentially for specific ranks (heterogeneous code), see the review in (Summerfield et al., 2020). Experimental evidence for the existence of a summation code was provided by (Ganguli et al., 2008; Fitzgerald et al., 2013; Rutishauser et al., 2018; Chafee, 2013), see section 6 of the review (Summerfield et al., 2020) and Box 3 in (Bottini and Doeller, 2020) for an overview. The summation code was explained in (Ganguli et al., 2008) from a more general perspective: The surprising dominance of a single dimension in the network dynamics of area LIP (lateral intraparietal cortex). This effect is consistent with a large body of experimental data on the representation of numbers and other types of magnitudes in the brain (Nieder and Dehaene, 2009; Roitman et al., 2012; Nieder, 2016; Lyons et al., 2016; Lorenzi et al., 2021; Luyckx et al., 2019; Summerfield et al., 2020; Bottini and Doeller, 2020; Viganò et al., 2024; Xu et al., 2024). Informally one refers to

these common types of 1-D neural representations as mental number line and a 1-D cognitive map (Bottini and Doeller, 2020). It is closely related to the theoretically postulated geometric mental line of (Di Antonio et al., 2024). One functional advantage of such a scalar neural network code is that downstream network can use the same decoder for many tasks (Fitzgerald et al., 2013).

The presence of a summation code for ranks of items in an order makes transitive inference easy: Downstream networks just have to compare the firing activity of these neurons for two different items in an order. In fact, this computational strategy automatically gives rise to the symbolic distance effect that is seen in human behavior, see the review in (Mannella and Pezzulo, 2024): Human subjects have lower reaction time and make fewer errors if the required transitive inference involves pairs of items whose ranks are substantially different. But a summation code does not explain the terminal item effect, whereby transitive inference that involves a terminal item in an order exhibits fewer errors and is carried out faster than for other pairs of items with the same rank difference.

It remains unknown by which learning process the underlying latent variable of an order, the rank of an item, is extracted from dispersed pieces of information about order relations between pairs of items and mapped to the largely 1-dimensional dynamics of area LIP. Previous models for learning order relations in the brain involved multi-layer perceptrons (Nelli et al., 2023) or recurrent neural networks (Lippl et al., 2024) that were trained through stochastic gradient descent, or involved a special-purpose neuromodulatory system that had been optimized for this task (Miconi and Kay, 2025). An exception is the approach of (Di Antonio et al., 2024), where also a shallow linear network is used for learning orders. They considered the case where all order relations between pairs of items that are presented during learning are guaranteed to be adjacent items in the order, and observed that learning of an order can be reduced in this case to solving a linear regression task. Achieving through learning a representation of ranks in neural networks with a basically 1-dimensional dynamics, like that of area LIP, has been addressed besides (Di Antonio et al., 2024) only with the multilayer neural network model of (Nelli et al., 2023), where the activation of neurons on a hidden layer exhibited a 1D dynamics after Backprop training.

We present a minimal model for learning to extract and represent ranks in area LIP that is not only consistent with a large body of experimental data, but is also supported by a rigorous mathematical theory. The simplest possible assumption about the underlying network architecture is to have direct synaptic connections between neural populations that represent items and a neural system that has an LIP-like 1-dimensional dynamics as in (Ganguli et al., 2008), or even a single linear neuron. We show that this simple architecture suffices for learning rank estimates, with a simple and biologically plausible rule for synaptic plasticity. Furthermore, this learning capability is supported by a rigorous mathematical theory, the Rank Convergence Theorem. In a second step we show that this simple model also enables rapid configuration of rank estimates in the light of new order information, for example if the relationship between two separately learned order relations becomes known, as considered in (Nelli et al., 2023). However, in contrast to their model, no separate learning of confidence values for all pairs of items is needed. Our simple model exploits the commonly observed fact that different objects and concepts tend to represented in higher brain areas by approximately orthogonal neural codes. For example, it was found that different concepts or items are represented in the medial temporal lobe (MTL) of the human brain through the firing of almost disjoint assemblies of concept cells (Fried, 2022), i.e., by approximately orthogonal high-dimensional (high-D) neural codes. Approximate orthogonality of neural codes was also found in the hippocampus of the rodent (Sun et al., 2025). Orthogonality of item representations, or even one-hot encoding, has also been commonly assumed in other models for order learning, see e.g. (Nelli et al., 2023).

In the last section of the paper we address another important result on neural representations of rank representations in the parietal cortex of the human brain. Although the primarily 1-dimensional dynamics of area LIP suggests that 2D projections of fMRI recordings of rank representations should lie on a line, they are consistently curved. The question why this is the case was explicitly raised in (Nelli et al., 2023). We provide a rigorous explanation of this effect based on the fact that ranks are not only represented in a summation code, but also by heterogeneous rank selective neurons. We show that this representation entails that 2D projections of neural codes for ranks lie on a curve, rather than on a line. The same theory also explains other questions about curved 2D projections of high-D neural codes that were addressed in (Okazawa et al., 2021).

2 Results

2.1 Architecture of a simple model for learning relations

Our goal is to understand how the brain area LIP, with its largely 1-D neural dynamics, can learn to extract rank estimates for some order "$\prec$" on items that are each encoded by some high-D vector in other higher brain areas that are synaptically connected to LIP. Information about this order "$\prec$" is typically provided in a piecemeal fashion, in the form of pieces of information $\mathbf{a}_i \prec \mathbf{a}_j$ that inform about the relation between two of the items. The learning goal of area LIP, to produce approximate rank estimates for all items, is non-trivial because the rank of an item $\mathbf{a}_i$ is a latent variable that depends on these pieces of information not only for order relations that involve this item $\mathbf{a}_i$, but also on the order relations between all other items.

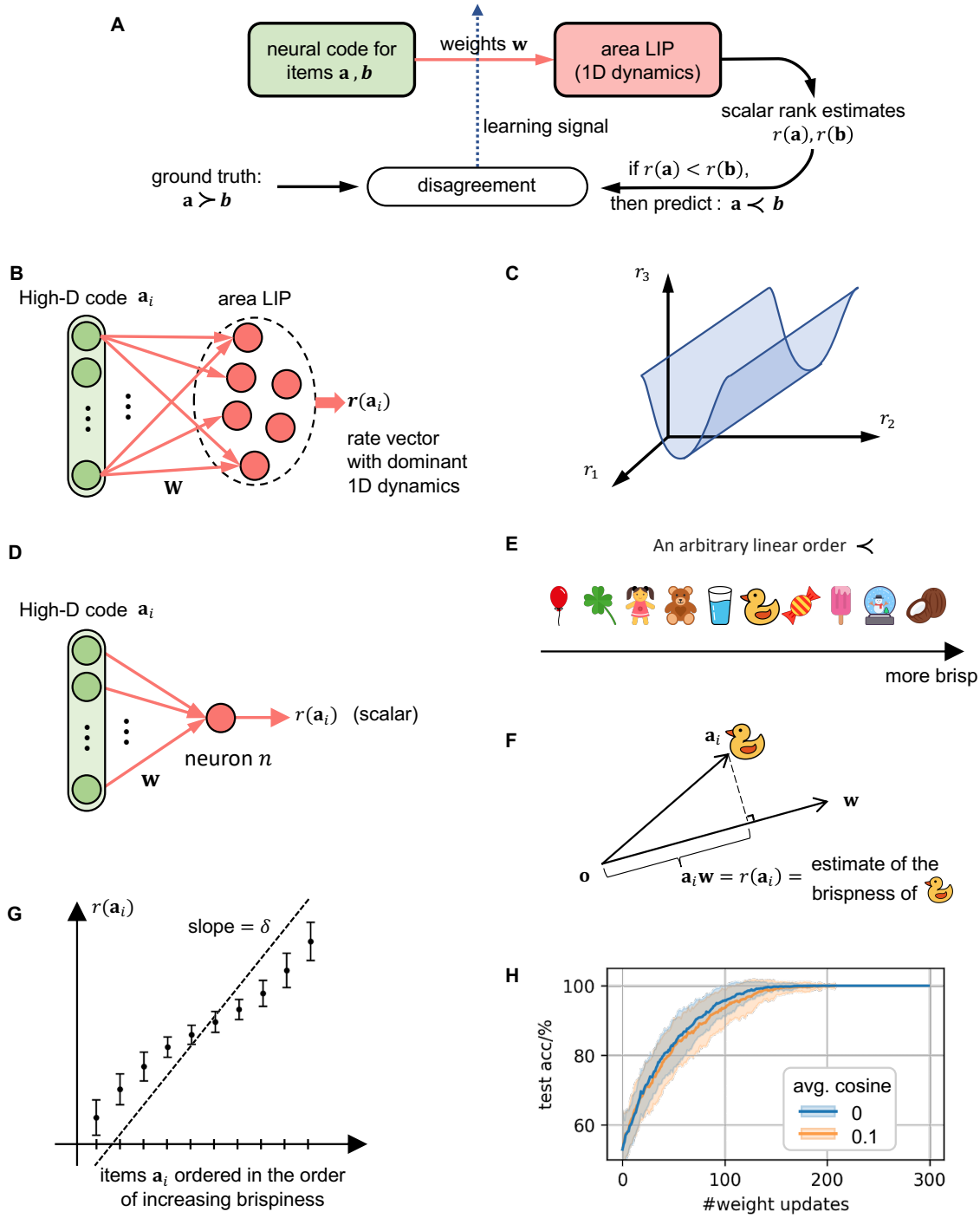
Thus area LIP learns rank estimates in a natural human setting through an online learning process, rather than through a batch learning process as in previous models. In online learning, weight updates are only carried out upon surprises, i.e., when the order between two items turns out to be different than expected on the basis of the rank estimates of the internal model that has been formed so far, see Fig. 1A. Our model can also be trained using batch learning, and the standard measure for evaluating learning speed—namely, the number of examples presented—yields roughly twice the number of weight updates observed in online learning, see section B.2 of the Supplement. We use in the following this standard measure for the speed of online learning. We show that this online learning process can be accomplished by a single layer of synapses with weights $\mathbf{w}$ to area LIP, as indicated in Fig. 1B. In other words, we show that after learning the overall firing activity $\mathbf{r}(\mathbf{a}_i)$ (space rate code) in a population of neurons that is targeted by these synaptic connections is able to provide consistent approximations of the ranks of all item $\mathbf{a}_i$ in a learned linear order "$\prec$".

According to the experimental data that we reviewed in the Introduction, the neural dynamics in area LIP is dominated by a 1-D line attractor (see Fig. 1C), that can for example be reproduced in a model through lateral connections between excitatory neurons (Ganguli et al., 2008). Hence, in order to make the learning process analytically tractable, we replace the population with 1-D dynamics of Fig. 1B in the simpler model of Fig. 1D by a single linear neuron n, that provides rank estimates through its firing rate $r(\mathbf{a}_i)$. The rank estimates by such a linear neuron model are not constrained to assume only positive values. But if this is desired, one can achieve it by adding a positive constant to the output of the linear neuron, that roughly corresponds to a non-zero baseline firing rate of a cortical neuron. The arguably simplest possible learning rule for the weights of this neuron takes the following form, that we term Rank Learning Rule:

> **Rank Learning Rule.** If information is received that $\mathbf{a}_i \prec \mathbf{a}_j$ for some items $\mathbf{a}_i$ and $\mathbf{a}_j$, but $\mathbf{w}\mathbf{a}_j$ is not by some given margin δ larger than $\mathbf{w}\mathbf{a}_i$, one updates $\mathbf{w}$ by adding a fraction η of $\mathbf{a}_j$ and subtracting a fraction η of $\mathbf{a}_i$:

$$\mathbf{w} \leftarrow \mathbf{w} + \eta \left(\mathbf{a}_j - \mathbf{a}_i\right). \tag{1}$$

In the Rank Learning Rule, δ and η are two arbitrary, fixed, positive values representing the margin and learning rate, respectively. In our experiment, the default values are $\delta = 1$ and $\eta = 0.1$.



This learning rule is arguably the simplest possible one that has a chance to perform well because it arises from gradient descent for the ranks of two fixed items $\mathbf{a}_i$ and $\mathbf{a}_j$, see section A of the Supplement. Hence it will eventually achieve that $r(\mathbf{a}_i) + \delta < r(\mathbf{a}_j)$ if it is sufficiently often iterated. But updates for this pair of items are in general interwoven with updates for other pairs, which may contain some of the same items or not. Hence it is not a-priori clear whether this simple learning rule converges, i.e., arrives at rank estimates that are consistent with all order information that has been received. For example, the weight updates might get caught in infinite loops, since an item $\mathbf{a}_i$ has to satisfy many relations $\mathbf{a}_i \prec \mathbf{a}_k$ and $\mathbf{a}_k \prec \mathbf{a}_i$ with other items $\mathbf{a}_k$. As long as the rank of $\mathbf{a}_i$ is not yet properly reflected in the output value of n for synaptic input $\mathbf{a}_i$, the Rank Learning Rule may sometimes add a fraction of $\mathbf{a}_i$ to $\mathbf{w}$ and sometimes subtract a fraction of it. These updates could potentially cancel each other. We show in Fig. 1E-F in an example that the Rank Learning

Fig. 1 *(previous page)*: **Architecture and performance of our model. A.** The underlying online learning framework: Weights are updated whenever the model produces through its rank estimates $r(\mathbf{a})$ and $r(\mathbf{b})$ predictions about the order between two items $\mathbf{a}$ and $\mathbf{b}$ that turn out to be wrong. **B.** Brain architecture for learning summation codes for rank estimates in area LIP: High-D representations of items are projected through synapses with weights $\mathbf{w}$ onto a population of neurons with approximately 1-D dynamics. **C.** Scheme for 1-D dynamics of firing rates in a network: Activity states of the network drawn to a 1-D attractor (compare with Fig. 2 of (Ganguli et al., 2008)). **D.** Analytically tractable version of the model in panel B, with a single linear neuron representing the firing activity of a population with 1-dimensional dynamics. **E.** Task illustration: Learning the relation "more brisp than" for 10 items. **F.** The brispiness prediction for an item $\mathbf{a}_i$ (here: a duck) by the linear neuron of panel D can be understood geometrically as the length of the projection of the high-D neural code for $\mathbf{a}_i$ onto the weight vector $\mathbf{w}$. Hence learning of $\mathbf{w}$ amounts to finding a direction of the weight vector where the lengths $r(\mathbf{a}_i)$ of all projections of neural codes for items $\mathbf{a}_i$ onto $\mathbf{w}$ is consistent with their brispiness relation. **G.** Brispiness predictions for all items were evaluated at the stage in the learning process when they become accurate (that takes on average 93.6 weight updates). The dots and error bars indicate the means and standard deviations, respectively, computed over 100 rounds of training using different random seeds. The resulting rank estimates exhibit a characteristic S-shape. **H.** Emergence of transitive inference capability in the model. The accuracy of the model was evaluated after each weight update on non-adjacent pairs of items, while the learner only received information about pairs of items that were adjacent in the learned order. Perfect orthogonality of item representations $\mathbf{a}_i$ was assumed for the blue curve, and approximate orthogonality with an average cosine of 0.1 between these vectors as assumed for the orange curve. One sees that the model does not require perfect orthogonality for fast learning. Means and standard deviations were calculated for 100 rounds of training using different random seeds.

Rule works very well for learning the relation "more brisp than" for 10 items, where "Brispiness" is a nonsensical relation (term borrowed from (Nelli et al., 2023)). For example, the item "candy" is more brisp than the item "yellow duck". The neuron learns from a stream of information that one item is more brisp than some other item, and updates its internal model for predicting brispiness when needed, as indicated in panel A. After, on average, 93.6 updates it produces the rank estimate of Fig. 1G that is consistent with all order information. Furthermore, Fig. 1H shows that these rank estimates enable immediately correct transitive inference for pairs of items that are not adjacent in this order.

Transitive inference can be carried out by downstream networks in a straightforward manner on the basis of rank estimates that are encoded by firing rates in area LIP: In order to determine which of two items comes later in the order they just have to evaluate which of them elicits a higher firing rate. This model is consistent with the commonly observed "symbolic distance effect" (Moyer and Bayer, 1976) in the performance of transitive inference by animals and human subjects: The accuracy of transitive inference increases, and the response time decreases, with the distance of two items in a learned order. Our model suggests that this effect arises because the comparison of two firing rates takes less time and is more reliable if the two firing rates are further apart.

The concrete rank estimates that emerge through the Rank Learning Rule, shown in Fig. 1G, provide an explanation for another commonly observed effect in transitive inference by human subjects: The "terminal effect" or "serial position effect" (Mannella and Pezzulo, 2024). This effect describes the fact that transitive inference for pairs of items in a learned order has higher accuracy and requires less response time, compared with other pairs of items that have the same rank distance, if one of the two items is an endpoint in the learned order. This effect can be explained by the S-shape of the firing rates that emerge through learning with the Rank Learning Rule according to Fig. 1G: Firing rates for items at the endpoints of the learned order relation are further apart from those for next-ranked items, compared with distances between firing rates for other pairs of items that have the same rank difference. This non-linearity arises from the fact that the rank estimates for items at the endpoints are only pushed towards the outside by the Rank Learning Rule, whereas items in the mid-range are subject to numerous conflicting updates that push them up and down.

The Rank Learning Rule works best if the items $\mathbf{a}_i$ are encoded by orthogonal vectors. Orthogonal codes have recently been found to arise commonly in the rodent hippocampus (Sun et al., 2025) But

experimental data suggest that visual objects and concepts are represented also in the human brain by approximately orthogonal neural codes: Through sparse assemblies of "concept cells" in the MTL (Fried, 2022). If one encodes these assemblies by binary vectors, where a "1" indicates that a neuron belongs to this assembly, these experimental data support a model where items are encoded by sparse high-D binary vectors. These vectors can not be assumed to be perfectly orthogonal, since assemblies for different concepts are known to have an overlap between 1% and 4%, depending on the degree by which the underlying concepts are associated (De Falco et al., 2016). Fig. S1A shows that this amount of overlap between assemblies amounts to a cosine similarity between 0.02 and 0.08 for the corresponding binary vectors. The orange curves in Fig. 1H, Fig. 2D-E, and Fig. S1B show that the learning performance is still very good for an average pairwise cosine value of 0.1 between vectors that represent different items. In other words, the level of orthogonality that is needed for the Rank Learning Rule is reached by the approximate orthogonality that has been found in neural codes for concepts in the MTL of the human brain, no matter whether they are associated or not.

It is not clear whether a transitive inference query "is $\mathbf{a}_i \prec \mathbf{a}_j$?" is processed by neural circuits of the brain by estimating the ranks of both items in a sequential manner, or in a fully parallel fashion by providing both items simultaneously as input to these neural circuits. Similarly, it is not clear whether order information $\mathbf{a}_i \prec \mathbf{a}_j$ is processed during learning by placing attention sequentially on each of the two items, or in a fully parallel manner. The sequential process can be supported by the argument that placing attention simultaneously onto two different items is difficult, at least without explicit training. Our model is neutral with regard to this question, since it can also be used for fully parallel computations: By giving neural codes for both items simultaneously as inputs to the linear neuron, but $\mathbf{a}_j$ with factor $+1$ and $\mathbf{a}_i$ with factor -1, as in the models of (Nelli et al., 2023; Di Antonio et al., 2024). After learning, the linear neuron will then output the rank difference of both items, and this value is positive if and only if the answer to the question "is $\mathbf{a}_i \prec \mathbf{a}_j$?" is "yes". Note that also the Rank Learning Rule is compatible with this fully parallel model, and the weight update is in this case a fraction of the total input $\mathbf{a}_j - \mathbf{a}_i$ that it receives.

Fig. 1G-H show that learning rank estimates for brispiness works very well in our simple model. But it is not a-priori clear whether this example is representative. A deeper analysis of the dynamics of the learning process is needed in order to determine whether it converges for any given order, and for arbitrary sequences of pieces of information that inform about the order relations between some pairs of items. This analysis is provided in the following section.

2.2 Learning theory for the Rank Learning Rule

A simple geometrical argument shows that there exists for any order and any encoding of items by orthogonal vectors a weight vector $\mathbf{w}$ which enables the neuron n to produce accurate rank estimates for all N items in the order. Fig. 1F suggests that it suffices to make sure that each item has a projection of a suitable length onto the vector $\mathbf{w}$. Hence one can simply define it for any correct set of ranks r_i for the items $\mathbf{a}_i$ as

$$\mathbf{w} = \sum_{i=1}^{N} r_i \mathbf{a}_i. \tag{2}$$

But this simple arguments, assumes that one has already correct rank estimates r_i, although these are not provided directly during the learning process. Furthermore, it does not entail that the Rank Learning Rule converges to a vector $\mathbf{w}$ that provides correct rank estimates. It had been shown in (Di Antonio et al., 2024) that for the special case where only pairs of items are presented during learning that are adjacent in the order that is to be learnt, learning of the order can be reduced to a linear regression task. The optimal solution can be approximated via gradient descent, where the gradient descent rule is identical to the Rank Learning Rule, except that it requires a varying continuous-valued factor that measures continuous-valued deviations from target values. In contrast, we show that the Rank Learning Rule reaches after finitely many steps an optimal solution with a fixed learning rate. Importantly, it learns an order also if non-adjacent pairs of items are presented during learning. This can not be achieved if order learning is formulated as regression task, because one would then have to provide the actual rank difference of the two items, which becomes only known after the order has been learnt.

The learning theory for the Rank Learning Rule is governed by a rigorous mathematical result, the Rank Convergence Theorem. This Theorem guarantees that the Rank Learning Rule converges

for every given total order "$\prec$" , and for any order of presentations of pairs of items during learning, after finitely many weight updates. The precise statement of the Rank Convergence Theorem (see Methods) is analogous to that of the famous Perceptron Convergence Theorem, which guarantees convergence of online learning with a simple plasticity rule (the delta learning rule) for learning binary classifications. But the Rank Learning Rule has to solve a more difficult task, since it does not receive target ranks by a teacher, which would correspond to the target classifications which the learner receives in the case of the Perceptron Convergence Theorem. Rather, it has to learn a latent variable, the rank, which has no analogue when one learns classifications. Consequently, one cannot adopt the proof the the Perceptron Convergence Theorem in order to prove the Rank Convergence Theorem. Rather, one needs a new argument for the proof. We present this argument in full detail in the Methods section.

Note that the Rank Convergence Theorem only guarantees that if one keeps on presenting pairs of items for which the estimated ranks are not yet consistent with their order relation, this process will end after finitely many steps with a weight vector $\mathbf{w}$ for which the rank estimates that neuron n provides are fully consistent with the order. But if one stops this learning process prematurely, the rank estimates that neuron n provides are in general not yet consistent with all order relations.

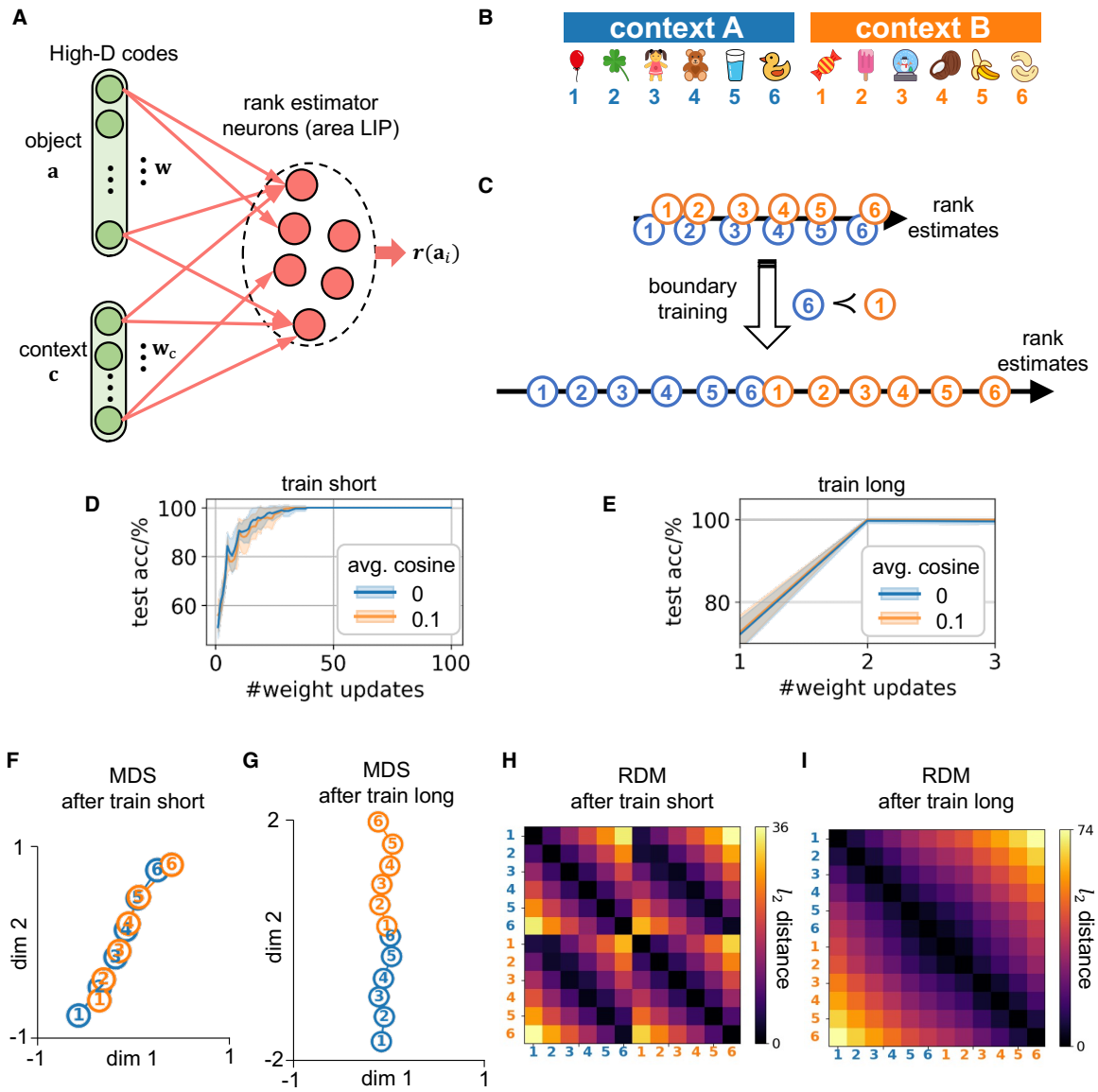
There may even be cases where the order relations between some pairs of items are unknown, e.g. when a partial order is to be learned. In this case the Rank Convergence Theorem guarantees that the learning process converges to rank estimates that are consistent with all pairwise order relations that exist in the partial order. In order to apply the Rank Convergence Theorem, which assumes existence of a total order that is consistent with all examples, one can use the fact that every partial order can be embedded into a total order.

An empirical finding is that the inference of new relations between items—those implied by other previously revealed relationships—reaches high accuracy quite fast (see Fig. 1H and Fig. 2D-E). The y-axes of these plots illustrate the emergent correctness of these inferences during learning. Furthermore, as detailed in Sec. 2.3, our model learns approximately 7–24× faster than humans in one experiment and 10× faster in another one under identical settings. This may be due to the presence of noise in neural circuits of the human brain, or to rules for synaptic plasticity that require more iterations. But one can adjust our learning model to these behavioral data by lowering its learning rate.

2.3 Rapid reconfiguration of the internal model when separately learned orders are combined on the basis of new information.

Humans can rapidly reassemble knowledge about separately learned orders, for example merge two separately learned orders on the basis of new information. This process was studied in experiments with human subjects in (Nelli et al., 2023). They first learned two separate orders in phase 1, and in phase 2 that the first item in one order comes after the last item in the other order, i.e., the learned orders were appended to each other, see Fig. 2B-C.

The Rank Convergence Theorem guarantees that the Rank Learning Rule will also lead in this 2-phase learning process to rank estimates that are in the end consistent with all order relations for the combined order. But it has no mechanism for fast generalization in the 2nd phase. We show here that if area LIP receives besides synaptic input that encodes items also synaptic input that encodes the contexts A and B to which the items of the two short orders belong (see Fig. 2A-B), then fast reconfiguration of the internal model becomes possible (Fig. 2D). It is well-known that higher brain areas, especially the PFC, keep track of the current context, and signal it via spike inputs to virtually all other brain areas (Heald et al., 2023). Furthermore, it is pointed out in that review that context is typically encoded in the PFC by firing patterns of neurons that are orthogonal to neural codes for other task-relevant variables (Bernardi et al., 2020). This was recently confirmed for the human brain in (Courellis et al., 2023).



In order to model learning of context-dependent ranks in area LIP we have complemented the Rank Learning Rule by a similarly simple Context Learning Rule for the weights of synaptic connections that project context information to area LIP:

Context Learning Rule.

If information is received that $\mathbf{a}_i \prec \mathbf{a}_j$ for some items $\mathbf{a}_i$ and $\mathbf{a}_j$ that come from two different contexts A and B and are encoded by neural codes $\mathbf{c}_A$ and $\mathbf{c}_B$, but $(\mathbf{w}\mathbf{a}_j + \mathbf{w}_c\mathbf{c}_B)$ is not by some given margin δ larger than $(\mathbf{w}\mathbf{a}_i + \mathbf{w}_c\mathbf{c}_A)$, one also updates the weights $\mathbf{w}_c$ for neural codes for context by adding a fraction η_c of $\mathbf{c}_B$ and subtracting a fraction η_c of $\mathbf{c}_A$:

$$\mathbf{w}_c \leftarrow \mathbf{w}_c + \eta_c(\mathbf{c}_B - \mathbf{c}_A). \tag{3}$$

We used random vectors $\mathbf{c}_A$ and $\mathbf{c}_B$ to encode contexts and applied a larger learning rate $\eta_c = 8 \times \eta$ to the context weights $\mathbf{w}_c$ compared to the other weights $\mathbf{w}$. Like η, the value of η_c is kept constant throughout the training process. We tested the resulting learning model for synaptic connections to area LIP on the scenario from (Nelli et al., 2023). First two different orders were learned in contexts A and B, see Fig. 2B, using the same weight vectors $\mathbf{w}$ and $\mathbf{w}_c$ for learning both orders. The learning curve is shown in 2D. For train long, we continued learning for these weight vectors with the pair of items that come last in the short order from context A and first in the other short order from context

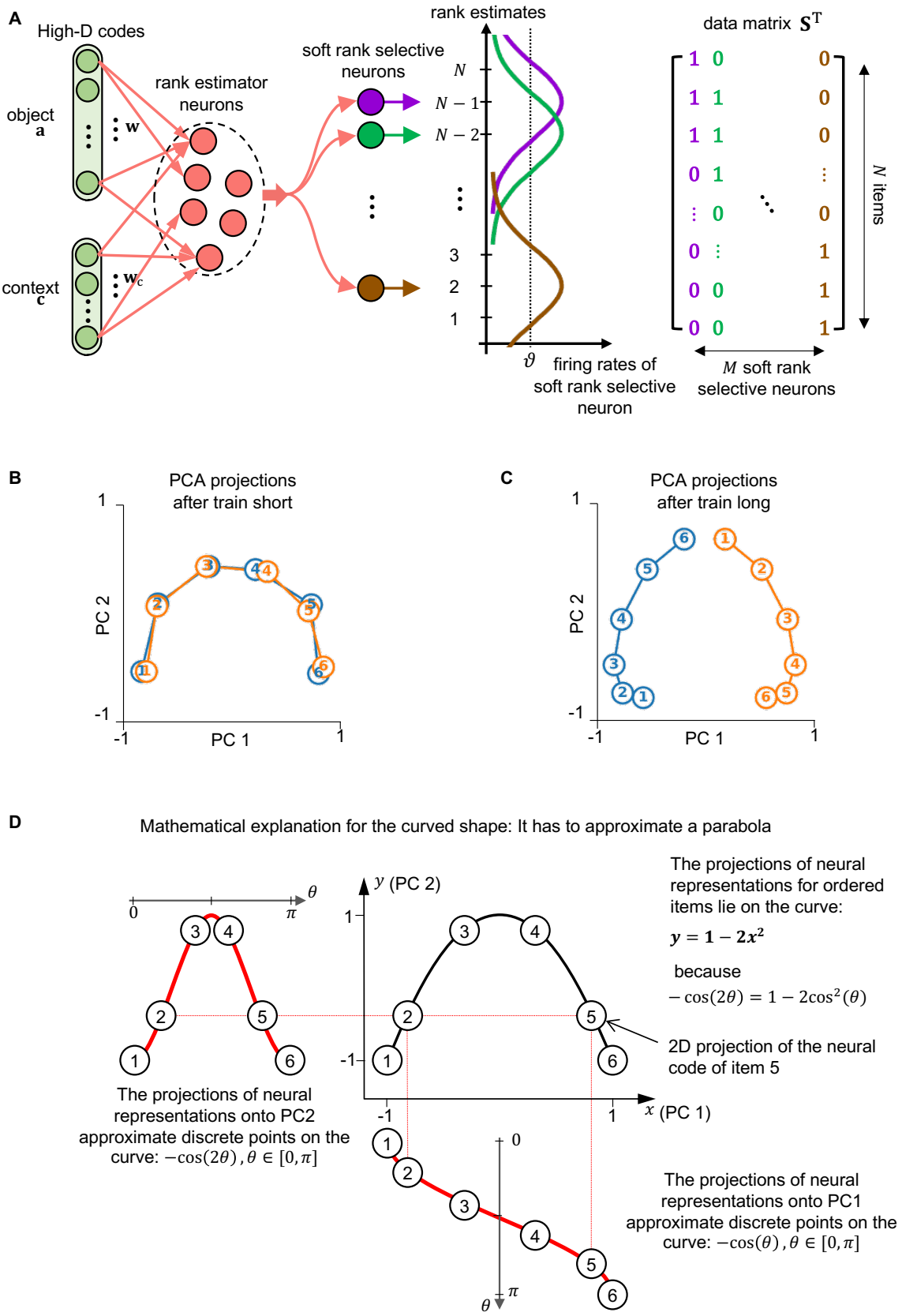
Fig. 2 *(previous page)*: **Combining information from separately learned orders A.** Network architecture: Rank estimation is computed as the sum of inputs from object coding (with weights $\mathbf{w}$) and context coding (with weights $\mathbf{w}_c$). **B.** Task illustration: Two linear orders are learned separately in different contexts A and B during "train short". **C.** In a 2nd learning phase "train long" the information is provided that the item with lowest rank in context B comes after the item with the highest rank in context A. Resulting rank estimates of our model after train short and train long are shown. **D.** Learning curve of the 1st learning phase "train short". Model accuracy was evaluated after each weight update on all item pairs within the same context, while the learner received information only about the adjacent pairs within each context. **E.** Learning curve of the 2nd learning phase "train long". Model accuracy was evaluated after each weight update on all item pairs in the combined order, while the learner received information only about the boundary pair. The blue curves in both **D** and **E** assume perfect orthogonality between item representations $\mathbf{a}$ and context representations $\mathbf{c}$, while the orange curves assume approximate orthogonality with an average cosine of 0.1. Mean and standard deviations in both **D** and **E** were computed from 100 training sessions with different random seeds across train-short and train-long conditions. The model learns rapidly without requiring perfect orthogonality, reaching 100% accuracy in all sessions after just two updates. **F.** and **G.** 2D projections via MDS (Multidimensional scaling) of the response of the rank estimator neurons from panel A for the items from the two contexts after train short and train long, respectively. **H.** and **I.** Representational dissimilarity matrices (RDMs) capture pairwise rank estimation distances as $RDM_{i,j} = \|\mathbf{r}(\mathbf{a}_i) - \mathbf{r}(\mathbf{a}_j)\|_2$ after short and long training, respectively. The same neural representation as for panels **F** and **G** was used.

B. The required number of examples for learning correct ranks for the combined order is shown in the learning curve of 2E. This enables then accurate transitive inference for the long order.

MDS projections of the representations of the items after train short and after train long are shown in Fig. 2F-G. Hence this simple model leads to very similar results as the model of (Nelli et al., 2023) for the same experiment, although the architecture and learning processes were very different in the two models. (Nelli et al., 2023) employed a multi-layer neural network that was trained through backpropagation (BP). Their model required in addition that on the side also a certainty matrix was learned for all pairs of items, that was not part of the neural network representation, but was needed for setting differential learning rates for different pairs of items. This was not needed in our model. Both models also differ in the learning speed: The simpler model requires fewer presentations of relations between items: 2 presentations of the boundary information were in general sufficient for learning the long order, see Fig. 2E.

2.4 Explaining the curved 2D projections of neural representations of ordered items in the human brain

One structural difference between the neural network representation of a learned order in the brain and in the multilayer perceptron (MLP) model of (Nelli et al., 2023) has been addressed in the Discussion section of that study: 2D projections of fMRI data from the human brain have a characteristic horseshoe shape, i.e., they are curved (inflected). A similar phenomenon had been observed in recordings from macaque PFC for 2D projections of brain representations of another scalar variable (Okazawa et al., 2021). In contrast, 2D projections of the MLP representations of items in a learned total order from (Nelli et al., 2023) tend to form a line. Fig. 2F-G show that this also occurs in our model. However, we have considered so far only 2D projections of responses of neurons that encode ranks by a summation code. The brain encodes ranks also by neurons that fire most strongly for particular ranks or regions of ranks (heterogeneous code) (Summerfield et al., 2020). We show here that the projection of their activity into 2D is necessarily horseshoe shaped since the projected points approximate a parabola, provided the neurons have some overlap in their preferred ranks.



Our model shows that neurons that encode ranks by a summation code emerge from a simple local learning rule. A transformation of this summation code into a heterogeneous code appears to be straightforward, and requires no new learning. If the outputs of summation coding neurons provide

Fig. 3 *(previous page)*: **Emergence of a parabola (horseshoe) shape of 2D PCA (principal component analysis) projections in the presence of soft rank selective neurons** **A.** Architecture for the generation of heterogeneous codes from a summation codes for rank estimates. Each colored circle in the middle indicates a population of excitatory neurons with negative feedback from inhibitory neurons, see Fig. S3 for a more detailed plot. This negative feedback curtails the firing rate in each population. On the other hand, staggered firing thresholds of excitatory neurons in different populations cause different onsets of their preferred ranks. Hence, the response profile of each population can be approximated by a Gaussian. If one binarizes their firing responses via some threshold, one arrives at the simple binary matrix on the right, which we analyze in Section 4.3 of Methods and Section E of the Supplement. **B.** and **C.** Projections of the patterns of firing activity of soft rank selective neurons for different items of an order onto the first two PCA dimensions, i.e., the first two eigenvectors of the covariance matrix. This is shown in panel **B** after training short, but before training long, and in panel **C** after training long. One sees that a characteristic horseshoe shape emerges in both cases. This curvature is consistent with 2D projections of BOLD signals from the human brain for items in an order (Nelli et al., 2023). **D.** According to Eq. (27) and (Eq. (28)), the projections of the firing activity of soft rank selective neurons onto the first principal component (PC1) approximate the cosine function over $[0, \pi]$, see the lower part of the plot, while the projections on PC2 approximate the cosine over $[0, 2\pi]$, see the plot at the left side. Combined, they come to lie on a parabola. This parabola explains why the horseshoe shape emerges in the PCA results of fMRI data. Note that we flipped the sign of both eigenvectors so that the PCA result resemble those in (Nelli et al., 2023).

synaptic inputs to excitatory neurons with diverse firing thresholds, these neurons will only fire for ranks above some minimal value. If these neurons interact in addition with inhibitory neurons so that a negative feedback loop arises, high ranks will activate this negative feedback so strongly that they effectively shut down responses to very high ranks. As a result, these excitatory neurons become tuned to some interval of ranks. The width of this tuning is regulated by the strength of inhibition: According to (Nieder, 2016) a blockage of GABAergic inhibition leads to a broadening of the tuning profiles of these excitatory neurons. Hence, a network architecture as indicated in Fig. 3A, see Fig. S3 for a more detailed plot, leads to a diversity of tuning profiles. Here the output of summation coding neurons is routed to different populations of excitatory neurons with staggered firing thresholds that each interact with a population of inhibitory neurons (both are enclosed by dashed lines in Fig. S3). This architecture creates a diversity of soft rank selective neurons with different approximately Gaussian tuning profiles in different populations, as indicated in the middle of Fig. 3A. The matrix on the right side of that panel records in each column the rank values for which one of these neurons is selective, in terms of an output firing rate being above some given threshold. Essential for our theoretical analysis is that these Gaussian tuning curves overlap, in this example by two rank values. Therefore we refer to them as soft rank selective neurons. The key point is that the covariance matrix that is generated by these soft rank selective neurons can be approximated by a circulant matrix, i.e., by a matrix where successive rows are shifted by one column, see Methods or (Gray et al., 2006) for a precise definition.

Our theoretical analysis of the reasons for horseshoe shaped 2D projections of neural activity is based on the structure of eigenvectors of circulant matrices: They can be approximated with the help of the cosine function, see section 4.3 of Methods. Of particular interest for the analysis of the first two dimension of a PCA projections are the projection of neural codes onto the first two eigenvectors, for which these approximations via the cosine function are shown in Fig. 3D. More precisely, projections of the neural responses for items onto the first eigenvector can be approximated by the cosine function over 180 degree. Furthermore, their projections onto the second eigenvector can be approximated by the cosine function over 360 degrees. Using the double-angle formula, $\cos(2\theta) = 2\cos^2(\theta) - 1$, one can express the shape of the PCA results in 2D with a parabolic function, which yields the horseshoe-shaped inflection in Fig. 3B and C. This provides a qualitative fit to the recorded data from the human brain, as reported in (Nelli et al., 2023). In Section 4.3 of Methods and Section E of the Supplement we provide a rigorous proof of this effect. Note that the signs of eigenvectors are flexible to choose, so the PCA results with the same figure flipped upside down or left to right are equally valid.

These panels also show that the approximation of the covariance matrix by a circulant matrix is already good enough for this small instance (with $N = 6$ items and $M = 8$ soft rank selective neurons) so that its eigenvectors of the covariance matrix behave qualitatively like those of a circulant matrix.

3 Discussion

We have shown that the all-important latent variable for learning an order, the rank of an item in this order, can be learned by a simple plasticity rule for direct synaptic connections from neurons that encode items or context to neurons in area LIP that participate in the commonly observed representation of learned ranks through a summation code (Summerfield et al., 2020; Bottini and Doeller, 2020), i.e., by a space rate code (see Fig. 1A). This latent variable has to be extracted from dispersed information about order relations between different pairs of items, and it is not a-priori clear whether this is possible. But the convergence of this extraction process is guaranteed by a new theoretical result, the Rank Convergence Theorem. The same online learning method may also explain the emergence of rank estimates in other brain areas such as the PFC (Brunamonti et al., 2016).

The emergent S-shape of resulting rank estimates in our model (Fig. 1G) provides a principled explanation for the commonly observed "terminal effect" or "serial position effect" in human behavior, whereby transitive inference performance is enhanced when the comparison involves items located at the end of a linear order, see (Mannella and Pezzulo, 2024) and its references. Using rank estimates for transitive inference also explains the commonly observed "symbolic distance effect", whereby transitive inference performance increases linearly with the distance of items in the learned order. This effect arises automatically when transitive inference is based on rank estimates that are encoded in an internal model by firing rates: If the firing rates that encode the rank estimates for two items have a larger distance, this difference can be detected faster and more reliably by downstream networks.

If one takes into account that LIP neurons also receive information about the context of the current relational learning process, one can explain by our model also rapid reconfigurations of rank estimates in the light of new order relations, as addressed in the experiments and model of (Nelli et al., 2023). An interesting difference between the learning process in their MLP model and our simple model is that our model does not require Backprop training, and also no differential regulation of learning rates for different pairs of items is required for reconfiguration of learned order representations. As a result, substantially fewer learning steps are needed by our model for the reconfiguration of learned order relations in the face of new information.

Our model for learning order relations gives rise to several experimentally testable predictions. One prediction concerns the connectivity: It predicts that approximately orthogonal neural representations of order items in other brain areas are projected by direct synaptic connections into LIP neurons that encode after learning estimates for the rank of order items through a summation code. Furthermore, our model predicts that the weights of these direct synaptic connections undergo synaptic plasticity during learning of an order. Furthermore, our model predicts that no batch learning or order relations is required. Rather, the brain applies online learning, and synaptic plasticity gets only activated when the order information is "surprising", i.e., not yet predicted by the internal model of the brain. The synaptic plasticity mechanism at the direct synaptic connections to LIP neurons is predicted to be gated primarily by gating signals that are caused by surprises, also by presynaptic activity, but less by postsynaptic activity. In addition, our model predicts that information about the behavioral context of the currently learned order is also transmitted by direct synaptic connections to LIP neurons, and that the weights of these synapses are subject to similar rules for synaptic plasticity. Finally, our model predicts that in the case where several orders are learned, the ranks of items in each order can be decoded by the same decoder from the firing activity of LIP neurons.

In the last part of the paper we have proposed a theoretically founded solution to an open problem that was addressed in (Nelli et al., 2023): 2D projections of internal representations of ranks in the parietal cortex of the human brain are not straight, as predicted by most models, but on a horseshoe shaped curve. We have shown that if one takes into account that ranks are not only represented by summation codes of neurons, but also by soft rank selective neurons that preferentially fire for a certain range of ranks, the experimentally observed inflection of 2D projections automatically

emerges. It results from the structure of the first two eigenvectors of circulant matrices that arise in the mathematical analysis of neurons with soft selectivity for ranks. In fact, this analysis is applicable also to neurons with soft selectivity for some other magnitude, thereby also providing a new and principled explanation for the experimentally observed curved manifolds for decision variables in the macaque brain (Okazawa et al., 2021). Our theory predicts that horseshoe shaped curvature will arise in all 2D projections of brain activity that represents value of some scalar variable, provided that there exist neurons with soft selectivity for values of this variable.

The Rank Learning Rule that emerged from our normative approach is consistent with experimental data on synaptic plasticity in the brain, especially with more recent data from in-vivo experiments, see the reviews (Magee and Grienberger, 2020) and (Chéreau et al., 2022). It has been shown there that synaptic plasticity in the living and behaving animal depends primarily on presynaptic firing and on a gating signal, and less on the timing of postsynaptic spikes as proposed by the STDP (Spike-timing-dependent plasticity) rule (Markram et al., 1997). More specifically, the Rank Learning Rule that we have examined requires LTP (long-term potentiation) in order to increase the rank estimate for a synaptic input $\mathbf{a}_j$ and LTD (long-term depression) for decreasing the rank estimate for some synaptic input $\mathbf{a}_i$. It had been shown in (Hong et al., 2022) that separate eligibility traces for LTP and LTD exist, and that their impact on weight updates can be selectively gated by neuromodulatory signals.

Altogether, we have provided a new theoretical foundation for understanding the learning and representation of orders in the brain, in particular in area LIP.

4 Methods

4.1 Details to the Rank Learning Rule in Section 2.1

Here we introduce a loss function whose gradient descent optimization exactly reproduces the Rank Learning Rule.

Consider a fixed pair of items represented by coding vectors $\mathbf{a}_i$ and $\mathbf{a}_j$, where $\mathbf{a}_i \prec \mathbf{a}_j$, with a margin $\delta > 0$ defining the minimal desired separation between their rank estimates. Let the firing rates (rank estimates) of the rank estimator neuron for these items be defined as $r(\mathbf{a}_i) = \mathbf{w}\mathbf{a}_i$ and $r(\mathbf{a}_j) = \mathbf{w}\mathbf{a}_j$.

We define a pairwise ranking loss function as follows:

$$L_{i,j} = \begin{cases} \delta - (r(\mathbf{a}_j) - r(\mathbf{a}_i)), & \text{if } (r(\mathbf{a}_j) - r(\mathbf{a}_i)) < \delta, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Minimizing this loss function (See eq. (4) in Supplementary Sec. A. for detail) via gradient descent leads precisely to the weight update given by the Rank Learning Rule:

$$\Delta\mathbf{w} = -\eta \frac{\partial L_{i,j}}{\partial \mathbf{w}} = \eta \cdot \underbrace{e}_{\text{gating}} \cdot \underbrace{(\mathbf{a}_j - \mathbf{a}_i)}_{\text{pre-synaptic}}, \quad \text{where} \quad e = \begin{cases} 1, & \text{if } (r(\mathbf{a}_j) - r(\mathbf{a}_i)) < \delta, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

4.2 Proof of the Rank Convergence Theorem discussed in Section 2.2

Rank Convergence Theorem:

Let $\mathbf{a}_1, \ldots, \mathbf{a}_K$ be some arbitrary finite set of non-zero orthogonal vectors, to which we refer as items. We consider updates of weights $\mathbf{w}$ under the Rank Learning Rule:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta(\mathbf{a}_j - \mathbf{a}_i), \tag{6}$$

for arbitrarily selected pairs of items $\mathbf{a}_j$ and $\mathbf{a}_i$ with the property that $\mathbf{a}_i \prec \mathbf{a}_j$ but $\mathbf{w}_t\mathbf{a}_j$ is not by some margin δ larger than $\mathbf{w}_t\mathbf{a}_i$. The factor η is assumed to be some arbitrary positive fixed value.

Then $\mathbf{w}_t$ converges to some finite value, independently of the order and frequency by which pairs of items $\mathbf{a}_j$ and $\mathbf{a}_i$ are selected for updates.

Proof:

Assume, for contradiction, that the claim is false, i.e., that the learning process does *not* terminate, and infinitely many updates are performed by the Rank Learning Rule.

We consider the subset of items

$$T = \left\{ \mathbf{a}_k \;\middle|\; \begin{array}{l} \text{there are infinitely many weight updates for pairs} \\ \text{of items where one of them is } \mathbf{a}_k \end{array} \right\}. \tag{7}$$

By assumption, T is nonempty. We pick an element $\mathbf{a}_{\min} \in T$ that is minimal under the ordering $\prec$.

Observation 1: $\mathbf{w}_t\mathbf{a}_{\min} \to -\infty$.

Since only finitely many updates are made for items $\mathbf{a}_k \prec \mathbf{a}_{\min}$, from some step t on, all weight updates that involve $\mathbf{a}_{\min}$ also involve some $\mathbf{a}_{\min} \prec \mathbf{a}_k$. Hence from some step t on only negative updates are made for $\mathbf{a}_{\min}$, i.e., $\eta\mathbf{a}_{\min}$ is subtracted from $\mathbf{w}$ whenever a weight update is made that involves $\mathbf{a}_{\min}$. Hence, the orthogonality of the items $\mathbf{a}_i$ implies that $\mathbf{w}_t\mathbf{a}_{\min} \to -\infty$.

Observation 2: A dual version of this argument shows that

$$\mathbf{w}_t\mathbf{a}_{\max} \to +\infty, \tag{8}$$

for the maximal item $\mathbf{a}_{\max}$ in T with regard to $\prec$.

Observation 3: Let item $\mathbf{a}_k$ in T have the property that $\mathbf{w}_t\mathbf{a}_k \to -\infty$ and $\mathbf{w}_t\mathbf{a}_l \to -\infty$ for all $\mathbf{a}_l$ in T with $\mathbf{a}_l \prec \mathbf{a}_k$. If $\mathbf{a}_q$ is the next larger item in T with regard to $\prec$, then also $\mathbf{w}_t\mathbf{a}_q \to -\infty$.

This holds because after a while, all updates of $\mathbf{w}$ where $\eta\mathbf{a}_q$ is added to $\mathbf{w}_t$ are made at steps t when $\mathbf{w}_t\mathbf{a}_q \leq \mathbf{w}_t\mathbf{a}_r + \delta$ for some $\mathbf{a}_r$ in T with $\mathbf{a}_r \prec \mathbf{a}_q$. Since by assumption $\mathbf{w}_t\mathbf{a}_r \to -\infty$ for all these $\mathbf{a}_r$, the products $\mathbf{w}_t\mathbf{a}_q$ that occur at these steps also converge to $-\infty$.

Induction along $\prec$, where the induction beginning is provided by Observation 1 and the induction step by Observation 3, implies then that $\mathbf{w}_t\mathbf{a}_k \to -\infty$ for all $\mathbf{a}_k$ in T. But this yields a contradiction to Observation 2.

This completes the proof of the Rank Convergence Theorem.

4.3 Theoretical analysis of the shape of 2D projections of the firing activity of soft rank selective neurons for items in an order, discussed in Section 2.4

We analyze the application of PCA for reducing the dimensionality of the representation of N items by M soft rank selective neurons from M to 2. The goal of this section is to derive a closed-form solution for the projection results.

The structure of this section follows the steps of computing PCA:

1. Calculation of the covariance matrix
2. Approximation and eigenvector calculation of the covariance matrix
3. Calculation of the projections of neural representations onto the first two principal components

4.3.1 Calculation of the covariance matrix

Assume that we have M soft rank selective neurons s_1 to s_M for items $\mathbf{a}_1$ to $\mathbf{a}_N$ in an order. We denote in the following the response of neuron s_i to item $\mathbf{a}_j$ as $\mathbf{s}_i(j)$.

We now consider the case where

$$\mathbf{s}_i[j] = \begin{cases} 1, & \text{if } j \in \{i, i+1, \ldots, i+O_v\} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

These are the columns in the example matrix $\mathbf{S}$ as shown in Fig. 3A.

In this equation, O_v (overlap) represents the number of items for which a soft rank-selective neuron fires beyond its central rank.

The responses of all M soft rank selective neurons to the N items is encoded by the matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_N \end{bmatrix}. \tag{10}$$

Firstly, we compute the mean vector of the responses of the soft rank selective neurons to different items:

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i. \tag{11}$$

We then compute the centered data matrix:

$$\mathbf{S}_c = \begin{bmatrix} \mathbf{s}_1^c \\ \mathbf{s}_2^c \\ \vdots \\ \mathbf{s}_N^c \end{bmatrix}, \tag{12}$$

where each response vector is centered by subtracting the mean vector:

$$\mathbf{s}_i^c = \mathbf{s}_i - \boldsymbol{\mu}. \tag{13}$$

The covariance matrix $\mathbf{C} \in \mathbb{R}^{M \times M}$ is defined by:

$$\mathbf{C} = \frac{1}{N-1} \mathbf{S}_c^\top \mathbf{S}_c. \tag{14}$$

The variance terms in the main diagonal of $\mathbf{C}$ are:

$$\mathbf{C}_{ii} = \frac{1}{N-1} \sum_{k=1}^{N} (\mathbf{s}_k^c[i])^2. \tag{15}$$

The covariance terms are:

$$\mathbf{C}_{i,j} = \frac{1}{N-1} \sum_{k=1}^{N} \mathbf{s}_k^c[i] \mathbf{s}_k^c[j]. \tag{16}$$

The principal components are the first two eigenvectors of $\mathbf{C}$. However, without simplification, it is difficult to obtain a closed-form solution. Hence, we will introduce some assumptions to simplify the equations in the following sections.

4.3.2 Approximation and eigenvector calculation of the covariance matrix

We show that $\mathbf{C}$ is a banded symmetric Toeplitz matrix, and hence can be approximated by a circulant matrix. Then we use properties of the eigenvectors of circulant matrices.

Assumption. Assume N is reasonably large so the edge effect can be neglected. Specifically, we assume that:

$$\forall j \in [1, M], \quad \text{we have} \quad \boldsymbol{\mu}[j] \approx \frac{O_v + 1}{N}. \tag{17}$$

This equation holds rigorously for items that are not at an end of an order.

The centered data vectors then can be approximated by:

$$s_i^c[j] \approx \begin{cases} 1 - \dfrac{O_v + 1}{N}, & \text{if } i \leq j \leq i + O_v \\ \\ -\dfrac{O_v + 1}{N}, & \text{otherwise.} \end{cases} \tag{18}$$

Since $\dfrac{O_v + 1}{N}$ is small for large N, we can further approximate:

$$s_i^c[j] \approx \begin{cases} 1, & \text{if } i \leq j \leq i + O_v \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

Under our assumption, the covariance matrix $\mathbf{C}$ is symmetric and Toeplitz, meaning that $\mathbf{C}_{i,j}$ depends only on $|i - j|$.

The covariance between dimensions i and j is:

$$\mathbf{C}_{i,j} = \frac{1}{N-1} \sum_{k=1}^{N} s_k^c[i] s_k^c[j]. \tag{20}$$

Given the approximated centered data, $s_k^c[i] s_k^c[j]$ is non-zero only when neural representations of items overlap in dimensions i and j. Hence, we can approximate $\mathbf{C}_{i,j}$ by:

$$\mathbf{C}_{i,j} = \begin{cases} (O_v + 1)/(N-1), & \text{if } |i - j| = 0 \\ O_v/(N-1), & \text{if } |i - j| = 1 \\ (O_v - 1)/(N-1), & \text{if } |i - j| = 2 \\ \vdots & \\ 0, & \text{if } |i - j| > O_v \end{cases}. \tag{21}$$

The form of the resulting covariance matrix is the **banded symmetric Toeplitz matrix** that satisfies the following properties:

- **Toeplitz structure:** The matrix has constant values along each diagonal, i.e., $\mathbf{C}_{i,j} = \mathbf{C}_{i+k,j+k}$ for all i, j, k such that the indices are within bounds.
- **Symmetry:** The matrix is symmetric, meaning $\mathbf{C}_{i,j} = \mathbf{C}_{j,i}$.
- **Banded structure:** The matrix has nonzero entries only within a fixed number of diagonals. Specifically, there exists a bandwidth $b \geq 0$ such that $\mathbf{C}_{i,j} = 0$ whenever $|i - j| > b$. We have $b = O_v$ in our setting as depict in Eq. (21).

An example $\mathbf{C}$ is provided below:

$$\mathbf{C} = \begin{pmatrix} c_0 & c_1 & \cdots & \cdots & \mathbf{0} \\ c_1 & c_0 & c_1 & \cdots & \vdots \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & c_0 & c_1 \\ \mathbf{0} & \cdots & \cdots & c_1 & c_0 \end{pmatrix}. \tag{22}$$

A banded symmetric Toeplitz covariance matrix can be approximated by a **circulant matrix** $\mathbf{C}' \in \mathbb{R}^{M \times M}$ (Gray et al., 2006), which, by definition, has a:

- **Cyclic row structure:** Each row of $\mathbf{C}'$ is obtained by shifting the preceding row one element to the right (wrapping around at the boundaries). If the first row is

$$(c'_0, c'_1, \ldots, c'_{n-1}),$$

then the i-th row is given by

$$(c'_{(-i) \bmod n}, c'_{(1-i) \bmod n}, \ldots, c'_{(n-1-i) \bmod n}).$$

If one approximates the example banded symmetric Toeplitz matrix $\mathbf{C}$ above by a circulant matrix $\mathbf{C}'$, then $\mathbf{C}'$ would have the following form:

$$\mathbf{C}' = \begin{pmatrix} c_0 & c_1 & 0 & \cdots & 0 & c_1 \\ c_1 & c_0 & c_1 & 0 & \cdots & 0 \\ 0 & c_1 & c_0 & \ddots & & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & & \ddots & c_0 & c_1 \\ c_1 & 0 & \cdots & 0 & c_1 & c_0 \end{pmatrix}. \tag{23}$$

The only difference compared to $\mathbf{C}$ is that $\mathbf{C}'$ requires an additional elements c_1 in the upper right and lower left corner to fulfill the circulant property. Using a circulant matrix to approximate a banded Toeplitz matrix becomes more accurate as the size of the matrix M grows and the bandwidth O_v becomes narrower.

Importantly, a circulant matrix has **eigenvectors that can be represented by the columns of the discrete Fourier transform (DFT) matrix** (Gray et al., 2006). In the case of a real symmetric circulant matrix, the DFT eigenvectors can be re-expressed in a real-valued form. Specifically, the discrete cosine transform (DCT) provides an orthogonal basis that yields a closed-form representation of the eigenvectors (Britanak et al., 2010):

$$\mathbf{u}_k[j] = \cos\left(\frac{\pi k(2j+1)}{2M}\right), \tag{24}$$

where $k = 1, 2, \ldots, M$ is the mode index, and $j = 1, 2, \ldots, M$ is the position index of each eigenvector. The proof of the validity of this step can be found in the supplementary information Sec. E.

We focus on the first two principal components, which correspond to the eigenvectors with the largest two eigenvalues (usually corresponding to $k = 1$ and $k = 2$):

- **First eigenvector:**

$$\mathbf{u}_1[j] = \cos\left(\frac{\pi(2j+1)}{2M}\right).$$

- **Second eigenvector:**

$$\mathbf{u}_2[j] = \cos\left(\frac{2\pi(2j+1)}{2M}\right).$$

4.3.3 Calculation of the projections of neural data onto the first two principal components

The projection of $\mathbf{s}_i^c$ onto the k-th principal component is:

$$(\mathbf{q}_i)_k = \mathbf{u}_k \mathbf{s}_i^c = \sum_{j=1}^{M} \mathbf{u}_k[j] \mathbf{s}_i^c[j]. \tag{25}$$

Substituting the approximated $\mathbf{u}_k[j]$ and $\mathbf{s}_i^c[j]$ from eq. (24) and eq. (19), we get:

$$(\mathbf{q}_i)_k = \sum_{j=i}^{i+O_v} \cos\left(\frac{\pi k(2j+1)}{2M}\right). \tag{26}$$

An intuitive observation is that the projection result can be viewed as a windowed sum of cosine values from i to $i + O_v$, where O_v (overlap) represents the number of items for which a soft rank-selective neuron fires beyond its central rank. With $k = 1$ and under the assumption that the window is short (i.e., small O_v), this windowed sum approximates a slightly smoothed cosine function over $[0, \pi]$ for $i \in [0, N]$ (Fig. 3D, x-axis). Therefore, one gets the projection onto PC1 as:

$$x(i) = \sum_{j=i}^{i+O_v} \cos\left(\frac{\pi(2j+1)}{2M}\right) \approx \cos(\theta), \quad \text{where} \quad \theta = \frac{\pi(2i+1)}{2M}. \tag{27}$$

Similarly, the projection onto PC2 ($k = 2$) follows a discrete cosine approximation over $[0, 2\pi]$ for $i \in [0, N]$ (Fig. 3D, y-axis):

$$y(i) = \sum_{j=i}^{i+O_v} \cos\left(\frac{2\pi(2j+1)}{2M}\right) \approx \cos(2\theta), \quad \text{where} \quad \theta = \frac{\pi(2i+1)}{2M} \text{ as above.} \tag{28}$$

With $(x, y) \approx (\cos(\theta), \cos(2\theta))$, where $\theta \in [0, \pi]$, and using the trigonometric identity $\cos(2\theta) = 2\cos^2(\theta) - 1$, one can derive the explicit shape of the PCA results as:

$$y = 2x^2 - 1, \tag{29}$$

which follows a parabolic curve.

In addition, the property of eigenvectors allows the sign of each eigenvector to be chosen freely. To better align with the PCA results in (Nelli et al., 2023), we flipped the sign of the projection result on both axes: $(x, y) \approx (-\cos(\theta), -\cos(2\theta))$ and get:

$$y = 1 - 2x^2, \tag{30}$$

which explains the inflection effect in Fig. 3D.

4.4 Details to learning a total order (Fig. 1)

In Fig. 1H, we trained our model on adjacent pairs among the 10 items, and tested on unseen non-adjacent pairs among the same set of items. The learning rate η in Eq. (5) is set to 0.1. Each item code $\mathbf{a}_i$ is a 10^4 dimensional random binary vector with number of ones uniformly sampled within the range $[0.2\%, 1\%]$. The blue curve is measured when all $\mathbf{a}_i$ forms an orthogonal set; and the orange curve is measured when the average cosine similarity between all pairs of items' code is 0.1. Each synaptic weight in $\mathbf{w}$ is initialized by a normal distribution $\mathcal{N}(0, 10^{-4})$. The weights are initialized very small to prevent unnecessary random biases that decreases the learning speed. The standard deviations of the two curves are calculated by repeated training from scratch 100 times. The randomness comes from data sampling, as a random pair of adjacent objects is picked at each training step. The margin δ is set to 1.

4.4.1 Method for generating codes for items with a given average pairwise cosine similarity

After initializing each item code with a fixed number of ones, this number is kept constant for each item. The indices are adjusted either to overlap with the dimensions of other items' ones, increasing the average cosine similarity, or to shift overlapping ones to different dimensions, reducing the cosine similarity. For a given target average cosine similarity, this process is iterated for a sufficiently long time to achieve a numerically optimal coding configuration.

4.5 Details to combining separately learned orders (Fig. 2)

4.5.1 Architecture

When generate the MDS plot in Fig. 2F-G, we use a population of rank estimator neurons $\mathbf{n}$'s activations to encode rank information collectively. The population in this experiment contains 100 rank estimator neurons. These rank estimator neurons receive summation inputs from both codes, with each neuron's value r:

$$r = \mathbf{w}\mathbf{a} + \mathbf{w}_c\mathbf{c}. \tag{31}$$

Each neuron learns independently its incoming synaptic weights based on the same learning rule Eq. (5) as a single rank estimator neuron. The symmetry breaks because of the different initialization with each synaptic weights sampled from $\mathcal{N}(0, 10^{-4})$.

4.5.2 Train short

During initial training ("train short"), we divided the 12 objects into two distinct sets (context A 1-6 and context B 1-6) and presented in alternating trails. The model only made comparisons within each context and indicate which of two adjacent-ranked objects (e.g., context A 3 and context A 4) was larger or smaller, receiving fully informative feedback. In total, the model is trained on 80 pairs of examples to form the results in Fig. 2F and H. This training allowed our model to infer ranks within each set (i.e., context A 1–6 and context B 1–6) but did not reveal any information about the rank relations between the two sets (e.g., context A 2 $<$ context B 3). We set the learning rate $\eta = 0.1$. Given that training occurs only between items from the same context, no weight updates to $\mathbf{w}_c$ would occur based on the context learning rule.

4.5.3 Train long

In (Nelli et al., 2023), human participants underwent 20 training trials (each contains a pair of object together with their ranking information) after being informed that the lowest ranked object "6" in context A is ranked lower than the lowest ranked object "1" in context B. We replicated this setting, and reproduced their results by training our model on this boundary pair for 2 times, only 1/10 of the total training trials of the human participants. The result is shown in Fig. 2G and I.

The key of assembling two order is to shift the rank estimation of a whole group together, without altering the in-context ranking, which, in our model, corresponds to prioritize the learning to happen on the context information than the item code. By analyze the contribution by different variables on the rank estimation changes after applying the weight updating in Eq. (5), with an assumption that codes are orthogonal, one gets:

$$\Delta r \propto \underbrace{\eta\|\mathbf{a}\|_2^2}_{\mathbf{a}'\text{s contribution}} + \underbrace{\eta_c\|\mathbf{c}\|_2^2}_{\mathbf{c}'\text{s contribution}} . \tag{32}$$

This result from our model suggests that two potential factors may contribute to the context-prior effect: 1. The learning rate for the context-related weights η_c is higher than the learning rate η for the object weights. 2. The l_2-norm of the context code $\mathbf{c}$ is larger than that of the object code $\mathbf{a}$. A detailed analysis of how these two factors influence the prioritization of learning based on context information is provided in the supplementary information Sec. C. In our experiment, we set $\eta_c = 0.4$ and $\eta = 0.05$; the dimension $D = 10^4$ of $\mathbf{a}$ equals to the dimension of $\mathbf{c}$; the sparsity of $\mathbf{a}$ equals to the sparsity of $\mathbf{c}$ with 0.2% to 1% (sampled from the uniform distribution) neurons fire in each code, which is based on the data from human medial temporal lobe (Fried, 2022; Waydo et al., 2006).

Acknowledgments

We gratefully acknowledge the generous advice of Christopher Summerfield, and comments by Giovanni Pezzulo, Alice Dauphin, and Lin Hui on an earlier version of the ms. This research was partially supported by the National Science Foundation of the USA (EFRI BRAID project 2318152) and the Austrian Science Fund (FWF) (10.55776/COE12).

References

Bernardi, S., M.K. Benna, M. Rigotti, J. Munuera, S. Fusi, and C.D. Salzman. 2020. The geometry of abstraction in the hippocampus and prefrontal cortex. *Cell 183*(4): 954–967 .

Bottini, R. and C.F. Doeller. 2020. Knowledge across reference frames: Cognitive maps and image spaces. *Trends in Cognitive Sciences 24*(8): 606–619 .

Britanak, V., P.C. Yip, and K.R. Rao. 2010. *Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations*. Elsevier.

Brunamonti, E., V. Mione, F. Di Bello, P. Pani, A. Genovesio, and S. Ferraina. 2016. Neuronal modulation in the prefrontal cortex in a transitive inference task: Evidence of neuronal correlates of mental schema management. *Journal of Neuroscience 36*(4): 1223–1236 .

Chafee, M.V. 2013. A scalar neural code for categories in parietal cortex: Representing cognitive variables as "more" or "less". *Neuron 77*(1): 7–9 .

Chéreau, R., L.E. Williams, T. Bawa, and A. Holtmaat 2022. Circuit mechanisms for cortical plasticity and learning. In *Seminars in cell & developmental biology*, Volume 125, pp. 68–75. Elsevier.

Courellis, H.S., J. Mixha, A.R. Cardenas, D. Kimmel, C.M. Reed, T.A. Valiante, C.D. Salzman, A.N. Mamelak, R. Adolphs, S. Fusi, et al. 2023. Abstract representations emerge in human hippocampal neurons during inference behavior. *bioRxiv*: 2023–11 .

De Falco, E., M.J. Ison, I. Fried, and R. Quian Quiroga. 2016. Long-term coding of personal and universal associations underlying the memory web in the human brain. *Nature communications 7*(1): 13408 .

Di Antonio, G., S. Raglio, and M. Mattia. 2024. A geometrical solution underlies general neural principle for serial ordering. *Nature Communications 15*(1): 8238 .

Fitzgerald, J.K., D.J. Freedman, A. Fanini, S. Bennur, J.I. Gold, and J.A. Assad. 2013. Biased associative representations in parietal cortex. *Neuron 77*(1): 180–191 .

Fried, I. 2022. Neurons as will and representation. *Nature Reviews Neuroscience 23*(2): 104–114 .

Ganguli, S., J.W. Bisley, J.D. Roitman, M.N. Shadlen, M.E. Goldberg, and K.D. Miller. 2008. One-dimensional dynamics of attention and decision making in lip. *Neuron 58*(1): 15–25 .

Gray, R.M. et al. 2006. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory 2*(3): 155–239 .

Heald, J.B., D.M. Wolpert, and M. Lengyel. 2023. The computational and neural bases of context-dependent learning. *Annual Review of Neuroscience 46*(1): 233–258 .

Hong, S.Z., L. Mesik, C.D. Grossman, J.Y. Cohen, B. Lee, D. Severin, H.K. Lee, J.W. Hell, and A. Kirkwood. 2022. Norepinephrine potentiates and serotonin depresses visual cortical responses by transforming eligibility traces. *Nature communications 13*(1): 3202 .

Lippl, S., K. Kay, G. Jensen, V.P. Ferrera, and L. Abbott. 2024. A mathematical theory of relational generalization in transitive inference. *Proceedings of the National Academy of Sciences 121*(28): e2314511121 .

Lorenzi, E., M. Perrino, and G. Vallortigara. 2021. Numerosities and other magnitudes in the brains: a comparative view. *Frontiers in Psychology* 12: 641994 .

Luyckx, F., H. Nili, B. Spitzer, and C. Summerfield. 2019. Neural structure mapping in human probabilistic reward learning. *elife* 8: e42816 .

Lyons, I.M., S.E. Vogel, and D. Ansari. 2016. On the ordinality of numbers: A review of neural and behavioral studies. *Progress in brain research* 227: 187–221 .

Magee, J.C. and C. Grienberger. 2020. Synaptic plasticity forms and functions. *Annual review of neuroscience 43*(1): 95–117 .

Mannella, F. and G. Pezzulo. 2024. Transitive inference as probabilistic preference learning. *Psychonomic Bulletin & Review*: 1–16 .

Markram, H., J. Lübke, M. Frotscher, and B. Sakmann. 1997. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science 275*(5297): 213–215 .

Miconi, T. and K. Kay. 2025. Neural mechanisms of relational learning and fast knowledge reassembly in plastic neural networks. *Nature Neuroscience*: 1–9 .

Moyer, R.S. and R.H. Bayer. 1976. Mental comparison and the symbolic distance effect. *Cognitive psychology 8*(2): 228–246 .

Nelli, S., L. Braun, T. Dumbalska, A. Saxe, and C. Summerfield. 2023. Neural knowledge assembly in humans and neural networks. *Neuron 111*(9): 1504–1516 .

Nieder, A. 2016. The neuronal code for number. *Nature Reviews Neuroscience 17*(6): 366–382 .

Nieder, A. and S. Dehaene. 2009. Representation of number in the brain. *Annual review of neuroscience 32*(1): 185–208 .

Okazawa, G., C.E. Hatch, A. Mancoo, C.K. Machens, and R. Kiani. 2021. Representational geometry of perceptual decisions in the monkey parietal cortex. *Cell 184*(14): 3748–3761 .

Roitman, J.D., E.M. Brannon, and M.L. Platt. 2012. Representation of numerosity in posterior parietal cortex. *Frontiers in Integrative Neuroscience* 6: 25 .

Rutishauser, U., T. Aflalo, E.R. Rosario, N. Pouratian, and R.A. Andersen. 2018. Single-neuron representation of memory strength and recognition confidence in left human posterior parietal cortex. *Neuron 97*(1): 209–220 .

Summerfield, C., F. Luyckx, and H. Sheahan. 2020. Structure learning and the posterior parietal cortex. *Progress in neurobiology* 184: 101717 .

Sun, W., J. Winnubst, M. Natrajan, C. Lai, K. Kajikawa, A. Bast, M. Michaelos, R. Gattoni, C. Stringer, D. Flickinger, et al. 2025. Learning produces an orthogonalized state machine in the hippocampus. *Nature*: 1–11 .

Viganò, S., R. Bayramova, C.F. Doeller, and R. Bottini. 2024. Spontaneous eye movements reflect the representational geometries of conceptual spaces. *Proceedings of the National Academy of Sciences 121*(17): e2403858121 .

Waydo, S., A. Kraskov, R.Q. Quiroga, I. Fried, and C. Koch. 2006. Sparse representation in the human medial temporal lobe. *Journal of Neuroscience 26*(40): 10232–10234 .

Xu, B., J. Wu, H. Xiao, T.F. Münte, and Z. Ye. 2024. Inferior parietal cortex represents relational structures for explicit transitive inference. *Cerebral Cortex 34*(4): bhae137 .

Supplementary Information for

A parsimonious model for learning order relations provides a principled explanation of diverse experimental data

A Rank Learning Rule as local gradient descent

We show that for a fixed pair of items $\mathbf{a}_i$ and $\mathbf{a}_j$ the Rank Learning Rule from Eq. (1) and Eq. (5) is equivalent to performing gradient descent on a corresponding loss function for these two items. This analysis suggests that the Rank Learning Rule makes sense, but makes no implication for the realistic scenario where many different pairs of items, some with overlap, are presented to the learner.

Given a pair of items $\mathbf{a}_i \prec \mathbf{a}_j$ with rank estimates $r(\mathbf{a}_i) = \mathbf{w}\mathbf{a}_i$ and $r(\mathbf{a}_j) = \mathbf{w}\mathbf{a}_j$, and a margin $\delta > 0$, we define the loss function as:

$$L_{i,j} = \begin{cases} \delta - (r(\mathbf{a}_j) - r(\mathbf{a}_i)), & \text{if } (r(\mathbf{a}_j) - r(\mathbf{a}_i)) < \delta, \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

When calculating the gradient with respect to $\mathbf{w}$, we obtain:

$$\begin{aligned} \frac{\partial L_{i,j}}{\partial \mathbf{w}} &= \frac{\partial L_{i,j}}{\partial (r(\mathbf{a}_j) - r(\mathbf{a}_i))} \frac{\partial (r(\mathbf{a}_j) - r(\mathbf{a}_i))}{\partial \mathbf{w}} \\ &= \begin{cases} -(\mathbf{a}_j - \mathbf{a}_i) & \text{if } (r(\mathbf{a}_j) - r(\mathbf{a}_i)) < \delta, \\ 0 & \text{otherwise.} \end{cases} \\ &= -e(\mathbf{a}_i - \mathbf{a}_j), \end{aligned} \tag{34}$$

where

$$e = \begin{cases} 1, & \text{if } (r(\mathbf{a}_j) - r(\mathbf{a}_i)) < \delta, \\ 0, & \text{otherwise,} \end{cases} \tag{35}$$

is the error signal.

Hence, applying gradient descent for a fixed pair of items results in an update rule that matches Eq. (5):

$$\Delta \mathbf{w} = -\eta \frac{\partial L_{i,j}}{\partial \mathbf{w}} = \eta e(\mathbf{a}_j - \mathbf{a}_i). \tag{36}$$

B Further details to Sec 2.1

B.1 Impact of relaxed orthogonality of neural codes for items

The brain represents items by assemblies of concept cells, that fire when this item is presented (or mentally considered) (De Falco et al., 2016). An important question for this study is to what extent these neural presentations of items can be viewed as being represented by orthogonal vectors. These assemblies are sparse (consisting of 0.2% to 1% of neurons in the human medial temporal lobe (Waydo et al., 2006; Fried, 2022)). But some neurons respond to more than one item. In other words, the assemblies have in general some overlap. The relative size of the overlap between two assemblies of concept cells for two different concepts has been estimated experimentally by the relative frequency that a neuron that responds to one concept also responds to the other concept. More precisely, the overlap between two assemblies is measured as % of neurons in their intersection relative to the size if the union of both assemblies.

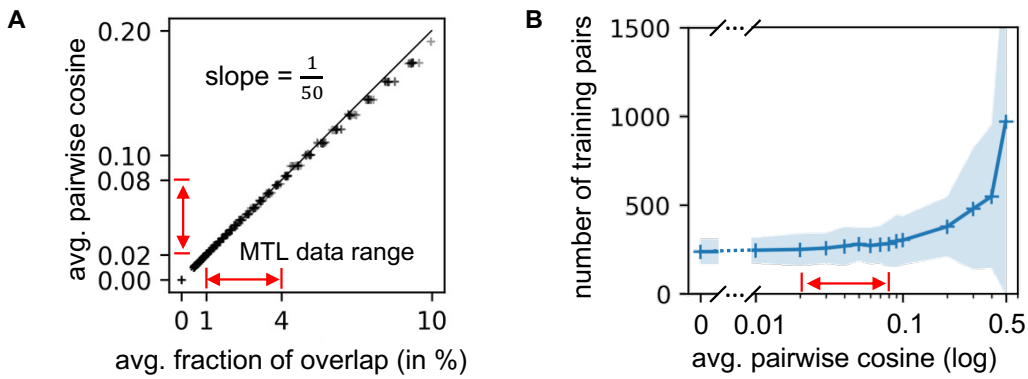


Fig. S1: **A.** The average fraction of overlap and the average pairwise cosine similarity follow a linear relationship in the range of small overlaps that we consider. Data from the human medial temporal lobe (MTL) suggest that the average fraction of overlap between different objects falls within the range of 1% to 4% (De Falco et al., 2016; Fried, 2022). From our simulation results, we observe that this range corresponds to an average cosine similarity between 0.02 and 0.08. A linear regression fit shows that the average cosine similarity is approximately $\frac{1}{50}$ of the average fraction of overlap, as depicted by the black line. **B.** Number of pairs of items presented, required for the agent to reach 100% testing accuracy with varying average cosine similarity, for the task of learning brispiness among 10 items. Lines and shaded areas represent the averages and standard deviations from 100 repeated experiments. Training is relatively fast when the average pairwise cosine is < 0.1, aligning with human MTL data $[0.02, 0.08]$. (Note: The x-axis of **B** is marked with (log), it means the scale of the axis adopts the logarithmic scale. Values shown on its side are still the original values.)

We show in Fig. S1A empirically that the fraction of overlap between assemblies for two different items is (in the range of small overlaps) approximately linearly related to the cosine similarity between the two corresponding binary vectors that encode these assemblies. This empirical analysis was carried out by assigning a binary vector of dimension 10^4 to each assembly (item), with sparseness uniformly sampled from the range of 0.2% to 1% of 1's. In addition, Fig. S1A shows that overlaps between 1 and 4 % yield cosine values between 0.02 and 0.08. The detailed method used to adjust the average fraction of overlap is discussed in Sec. 4.4.1.

Subsequently, we show in Fig. S1B that the learning speed of our model is for the range of these cosine values between 0.02 and 0.08 almost the same as for perfect orthogonality, i.e. cosine value 0. The task is the same as in Fig. 1E, where a single total order is learned. All settings are the same as depicted in Sec. 4.4.

B.2 Relationship between the number of weight updates in an online learning model and the number of item pairs needed to achieve the same performance in batch training using the rank learning rule

We illustrated the standard online learning approach in Fig. 1A: weights are updated only when there is a "surprise"; otherwise, they remain unchanged. For this online learning rule, we use the number of weight updates (in Fig. 1H and Fig. 2D-E) throughout the paper to evaluate learning speed. However, this measure for online learning differs from the standard measure in batch learning, where one counts the number of training examples (each example consists of a pair of items being compared). In this section we examine empirically the relationship between these two different measures for learning speed for the "briskness" learning experiment.

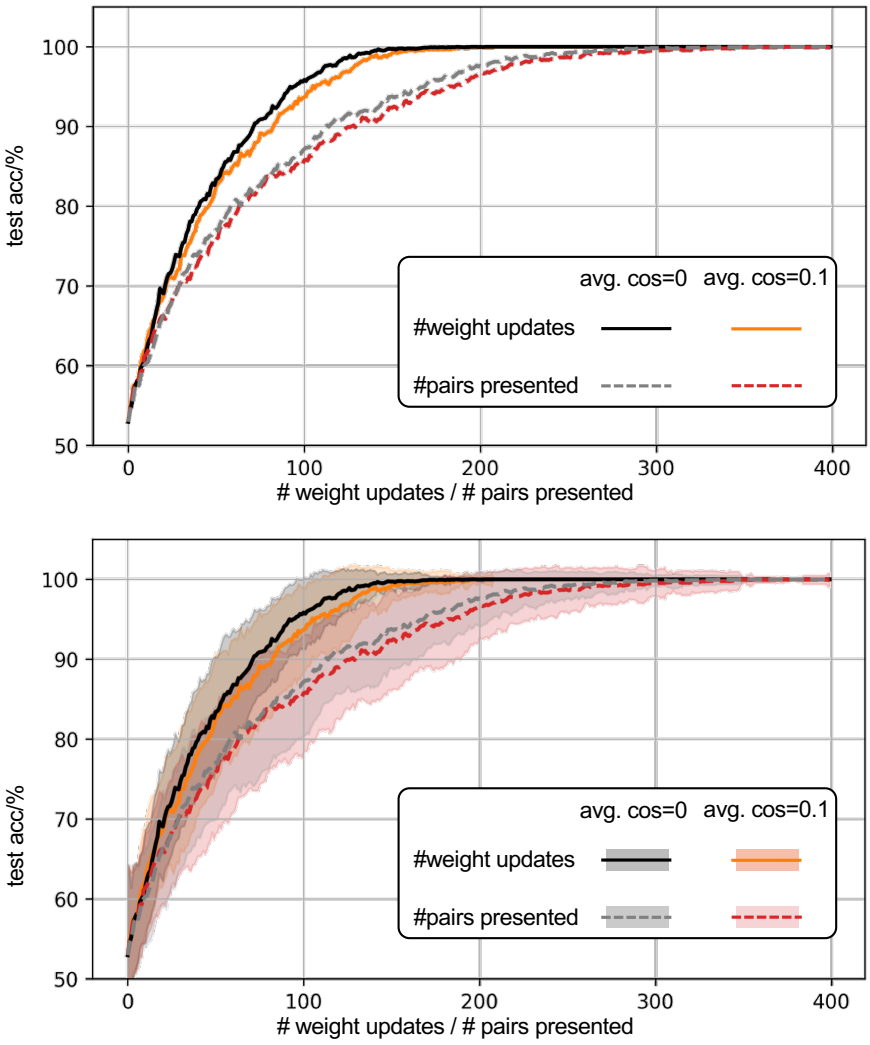


Fig. S2: Relationship between the number of weight updates and the number of pairs presented (standard deviation omitted in the upper panel for better visual clarity). The pairs are accompanied by their relative ranking information, which indicates which item is ranked higher. The comparison is made under conditions of both perfect orthogonality and approximate orthogonality (with an average cosine similarity of 0.1).

From the figure above, it is clear that there is a difference between the two measures. During training from scratch until the results become accurate, only about 55% of the presented pairs trigger weight updates in online learning, meaning that roughly 45% of the pairs do not lead to any update, even when all pairs are presented with equal probability. This indicates that, throughout the training process until accuracy is achieved, weight updates occur for only slightly more than half of the training pairs.

C Further details to Section 2.3

To understand how learning prioritizes the context code $\mathbf{c}$ over the item code $\mathbf{a}$, we analyze how weight updates affect rank estimates.

$$r = \mathbf{w}\mathbf{a} + \mathbf{w}_c\mathbf{c}. \tag{37}$$

After updating the weights according to Eq. (5), and considering different learning rates, we have:

$$\Delta\mathbf{w} = \eta\ e\ \left(\mathbf{a}_j - \mathbf{a}_i\right), \quad \Delta\mathbf{w}_c = \eta_c\ e\ \left(\mathbf{c}_j - \mathbf{c}_i\right). \tag{38}$$

When the compared objects $\mathbf{a}_i$ and $\mathbf{a}_j$ are from the same context, the second term is zero, indicating that $\mathbf{w}_c$ does not participate in the "train short" phase.

The new rank estimate of objects $\mathbf{a}_i$ and $\mathbf{a}_j$ are:

$$\begin{aligned} r'_i &= [\mathbf{w} + \eta\ e\ (\mathbf{a}_j - \mathbf{a}_i)]\mathbf{a}_i + [\mathbf{w}_c + \eta_c\ e\ (\mathbf{c}_j - \mathbf{c}_i)]\mathbf{c}_i, \\ r'_j &= [\mathbf{w} + \eta\ e\ (\mathbf{a}_j - \mathbf{a}_i)]\mathbf{a}_j + [\mathbf{w}_c + \eta_c\ e\ (\mathbf{c}_j - \mathbf{c}_i)]\mathbf{c}_j. \end{aligned} \tag{39}$$

Because different brain codes are typically orthogonal to each other:

$$\mathbf{a}_i\mathbf{a}_j = \begin{cases} \|\mathbf{a}_i\|_2^2, \text{ if } i = j, \\ \approx 0, \text{otherwise.} \end{cases}, \quad \mathbf{c}_i\mathbf{c}_j = \begin{cases} \|\mathbf{c}_i\|_2^2, \text{ if } i = j, \\ \approx 0, \text{otherwise.} \end{cases} \tag{40}$$

Therefore:

$$\begin{aligned} r'_i &\approx r_i - \eta\ e\ \|\mathbf{a}_i\|_2^2 - \eta_c\ e\ \|\mathbf{c}_i\|_2^2, \\ r'_j &\approx r_j + \eta\ e\ \|\mathbf{a}_j\|_2^2 + \eta_c\ e\ \|\mathbf{c}_j\|_2^2. \end{aligned} \tag{41}$$

When calculating the rank estimate change, the factors affecting the rank estimate changes are:

$$\Delta r \propto \left((\underbrace{\eta\|\mathbf{a}\|_2^2}_{\mathbf{a}'\text{s contribution}} + \underbrace{\eta_c\|\mathbf{c}\|_2^2}_{\mathbf{c}'\text{s contribution}})e \right). \tag{42}$$

The above theoretical analysis clarifies the methods available for prioritizing learning in different representations. Whether learning happens more on the object embedding or context embedding depends on the ratio between:

$$\left(\eta\|\mathbf{a}\|_2^2\right) \text{ v.s. } \left(\eta_c\|\mathbf{c}\|_2^2\right). \tag{43}$$

D Neural network architecture that leads to the emergence of soft rank selective neurons, see Section 2.4

We show here the architecture that was indicated in Fig. 3A in more detail.

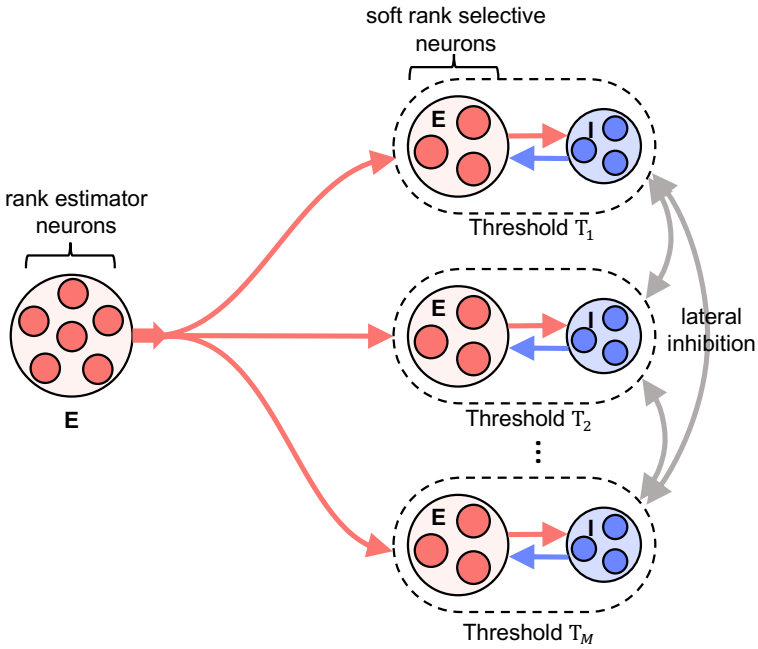


Fig. S3: A population of rank estimator neurons (summation coding neurons) projects to various populations of other neurons. Each population consists of interconnected excitatory and inhibitory neurons, with distinct firing thresholds for excitatory neurons in different populations. Inputs below the threshold value for a population do not activate it. When the inputs exceed the threshold, excitatory neurons are activated, but their firing activity is curtailed by negative feedback from inhibitory neurons. Inhibitory neurons may also provide lateral inhibition between populations in order to ensure that the overlaps in their tuning curves are limited.

E Proof of Validity for the Eigenvector Approximations in Sec. 4.3

We need to show that:

$$(\mathbf{C}\mathbf{u}_k)[i] = \lambda_k \mathbf{u}_k[i]. \tag{44}$$

Using the Toeplitz structure:

$$(\mathbf{C}\mathbf{u}_k)[i] = \sum_{j=1}^{M} \mathbf{C}_{ij}\mathbf{u}_k[j] = \sum_{n=1}^{M} c(|i-n|)\mathbf{u}_k[n]. \tag{45}$$

Substitute $\mathbf{u}_k[n]$:

$$(\mathbf{C}\mathbf{u}_k)[i] = \sum_{n=1}^{M} c(|i-n|)\cos\left(\frac{\pi k(2n+1)}{2M}\right). \tag{46}$$

Let $m = n - i$. Then $n = i + m$, and m ranges from $-(i-1)$ to $M - i$. Substituting:

$$(\mathbf{C}\mathbf{u}_k)[i] = \sum_{m=-(i-1)}^{M-i} c(|m|)\cos\left(\frac{\pi k(2(i+m)+1)}{2M}\right). \tag{47}$$

Use the Trigonometric identity:

$$\cos(A+B) = \cos A\cos B - \sin A\sin B. \tag{48}$$

Set $A = \frac{\pi k(2i+1)}{2M}$ and $B = \frac{\pi k(2m)}{2M}$, so:

$$\begin{aligned}\cos\left(\frac{\pi k(2(i+m)+1)}{2M}\right) &= \cos\left(\frac{\pi k(2i+1)}{2M}\right)\cos\left(\frac{\pi k(2m)}{2M}\right)\\ &\quad - \sin\left(\frac{\pi k(2i+1)}{2M}\right)\sin\left(\frac{\pi k(2m)}{2M}\right).\end{aligned} \tag{49}$$

Substitute this back:

$$\begin{aligned}(\mathbf{C}\mathbf{u}_k)[i] &= \cos\left(\frac{\pi k(2i+1)}{2M}\right)\sum_{m=-(i-1)}^{M-i} c(|m|)\cos\left(\frac{\pi k(2m)}{2M}\right)\\ &\quad - \sin\left(\frac{\pi k(2i+1)}{2M}\right)\sum_{m=-(i-1)}^{M-i} c(|m|)\sin\left(\frac{\pi k(2m)}{2M}\right).\end{aligned} \tag{50}$$

Since $c(|m|)$ is an even function:

- $\cos\left(\frac{\pi k(2m)}{2M}\right)$ is even, so the sum involving it is nonzero.
- $\sin\left(\frac{\pi k(2m)}{2M}\right)$ is odd, so the sum involving it is zero:

$$\sum_{m=-(i-1)}^{M-i} c(|m|)\sin\left(\frac{\pi k(2m)}{2M}\right) = 0. \tag{51}$$

Thus:

$$(\mathbf{C}\mathbf{u}_k)[i] = \cos\left(\frac{\pi k(2i+1)}{2M}\right)\sum_{m=-(i-1)}^{M-i} c(|m|)\cos\left(\frac{\pi k(2m)}{2M}\right). \tag{52}$$

Assume M is large and i is not near the edges. Approximate the sum over all m from $-\infty$ to ∞:

$$\sum_{m=-\infty}^{\infty} c(|m|)\cos\left(\frac{\pi k(2m)}{2M}\right) = \lambda_k. \tag{53}$$

Let:

$$\lambda_k = \sum_{m=-\infty}^{\infty} c(|m|) \cos\left(\frac{\pi k(2m)}{2M}\right). \tag{54}$$

We have:

$$(\mathbf{C}\mathbf{u}_k)[i] = \lambda_k \mathbf{u}_k[i]. \tag{55}$$

This shows that $\mathbf{u}_k$ is an eigenvector of $\mathbf{C}$, with eigenvalue λ_k.