LEARNING NEURAL REPRESENTATIONS IN TASK-SWITCHING GUIDED BY CONTEXT BIASES

Santiago Galella
Research Institute for Integrated Management of Coastal Areas,
Universitat Politècnica de València
Gandia, Spain
sgalella@upv.es

Salva Ardid
Research Institute for Integrated Management of Coastal Areas, and Department of Applied Physics,
Universitat Politècnica de València
Gandia, Spain
sardid@upv.es

ABSTRACT

The ability to switch between tasks effectively in response to external stimuli is a hallmark of cognitive control. Our brain can filter and integrate external information to accomplish goal-directed behavior. Task switching occurs rapidly and efficiently, allowing us to perform multiple tasks with ease. Similarly, artificial neural networks can be tailored to exhibit multi-task capabilities and achieve high performance across domains. In terms of explainability, understanding how neural networks make predictions is crucial for their deployment in many real-world scenarios. In this study, we delve into neural representations learned by *task-switching* networks, which use task-specific bias for multitasking. Task-specific biases, mediated by *context inputs*, are learned by alternating the tasks the neural network learns during training. By using the MNIST dataset and binary tasks, we find that *task-switching* networks produce representations that resemble other multitasking paradigms, namely *parallel* networks in the early stages of processing and *sequential* networks in the last stages, respectively. We analyze the importance of inserting task contexts in different stages of processing and its role in aligning the task with relevant features. Moreover, we visualize how networks generalize neural representations during *task-switching* for different tasks. The use of *context inputs* improves the interpretability of simple neural networks for multitasking, helping to pave the way for the future study of architectures and tasks of higher complexity.

Keywords Learning Representations · Neural Geometry · Context-Dependent Decision Making · Attention Mechanisms

Living involves constantly gauging and selecting the optimal action to perform. This decision is the result of the interaction of different elements, such as our current goals, external circumstances, or contexts (Monsell, 2003). In the brain, task switching and cognitive control have been associated principally with the prefrontal cortex (PFC), which provides top-down regulation to other cortical brain areas (Johnston et al., 2007). In this sense, PFC controls the activation of multiple neural pathways, so each is activated, or not, on demand during task execution (Menon & D'Esposito, 2022). Neural pathways are formed by collections of neurons that cooperate to yield a specific effect and it is believed they may conform the functional processing unit of the nervous system (Yuste, 2015; Saxena & Cunningham, 2019). For instance, in the visual cortex, subpopulations of neurons are responsible for the processing of different visual field features during object recognition (DiCarlo et al., 2012).

According to the gating theory (Miller & Cohen, 2001), context-dependent top-down inputs from the PFC regulate the activation of neural pathways. The encoding of relationships between stimulus and contexts

can be facilitated by tuning neural activity to multiple tasks. In the PFC of monkeys, it has been observed that single neurons exhibit nonlinear responses to multiple stimuli (Rigotti et al., 2013). This behavior, or mixed selectivity, produces a higher-dimensionality representation of the entire neural responses, which favors downstream neurons to easily group and classify task-dependent neural activity (Fusi et al., 2016).

Many models have been proposed to explain how *task-switching* is mediated in the brain (Cohen et al., 1990; Botvinick et al., 2001; Rougier & O'Reilly, 2002; Ardid & Wang, 2013). Recently, artificial neural networks have been revisited as models of neural computation, many findings suggesting their practicality for assessing brain theories (Richards et al., 2019). For example, recurrent neural networks with contexts have been used as models to study cognitive control in both humans and machines (Mante et al., 2013; Yang et al., 2019).

In the brain, neural representations refer to patterns of activity that stand for information processing, such as external features from the environment (Vilarroya, 2017). Understanding neural representations may improve the interpretability of intelligent systems based on neural networks. In the multi-task learning setting, we define *neural representations* as activity patterns that encode task-related processing. If neural ensembles have similar patterns of activation for different tasks, then we refer to these representations as *shared* representations. Previous work has also studied neural representations learned in artificial and biological neural networks, bridging the gap between brains and machines (Ito & Murray, 2021; Ito et al., 2022; Flesch, Juechems, et al., 2022). Methods to study neural representations in multi-task learning include population analysis tools to investigate how neural computations are associated with task-related stimulus features (Kriegeskorte et al., 2008; Jazayeri & Ostojic, 2021).

Multi-task learning can be accomplished through different network architectures. For example, we can use *sequential* networks to process each task independently. Single networks suffer from catastrophic forgetting when imposing moving from one task to another and back, but this can be avoided if each task-associated network is disconnected from others. This contrasts with *parallel* networks that process all tasks simultaneously. These networks benefit from learning shared representations, which regularizes and promotes generalization (Caruana, 1997), improving network performance across domains (Ruder, 2017). However, neural representations become entangled because neurons encode patterns of mixed selectivity. Identifying task-specific processing from these patterns is challenging. Also, what these patterns are, or serve, is yet to be characterized in terms of neural representations.

In this study, we introduce *task-switching* networks using *context inputs*, which diminish the entanglement of neural representations, similarly to *sequential* networks, while benefiting from the generalization of *parallel* networks. Similar ideas of using tasks biases have been previously reported in shallow networks to study multi-task efficiency (Musslick et al., 2017; Ravi et al., 2020) and to compare brain activity with neural geometry and dimensionality in neural networks (Flesch, Juechems, et al., 2022). Additionally, they have been used for continual learning and to prevent catastrophic forgetting (Masse et al., 2018; Grewal et al., 2021). To analyze learning in *task-switching* networks, we use *context inputs* to drive neural activity to perform different tasks. *Task-swiching* neural networks learn different biases for different tasks throughout the network depending on the *context input*. This executive control resembles that of top-down gating in the brain, mediated by the PFC.

We first investigate and compare the representations learned by *sequential*, *parallel* and *task-switching* networks (see results section *Task-switching* networks operate in a midway regime between *sequential* and *parallel* networks. We observe that *task-switching* networks produce representations that resemble *parallel* networks in the early stages of processing, whereas they resemble *sequential* networks in the latter stages. We introduce a metric to describe quantitatively shared representations learned by networks in multi-task by making use of representational dissimilarity matrices (RDMs) (Kriegeskorte et al., 2008), and show that *task-switching* networks potentiate shared representations further than *parallel* networks. We also show that *sequential* and *task-switching* in contrast to *parallel* networks underlie neural representations that disentangle the reconstruction of the task, increasing the interpretability of the model. Remarkably, *sequential* and *parallel* networks can be considered special cases of *task-switching* networks (see section *Task-switching* networks perform multiple tasks by extracting task congruency information). Moreover, we analyze the impact of context location and magnitude at different stages of the neural processing. The results of this analysis show that contexts introduced in the intermediate layers drive neural activity more strongly (section Context strength is maximized at intermediate layers), and show that the *task-switching* network is able to generalize task biases for unseen magnitudes of the context input (section *Task-switching* networks generalize representations between tasks). Finally, we expand previous analyses involving multi-task learning with contexts and mixed selectivity, showing that *task-switching* networks produce neural selectivity to multiple tasks (results section Mixed selectivity emerges for resolving task execution).

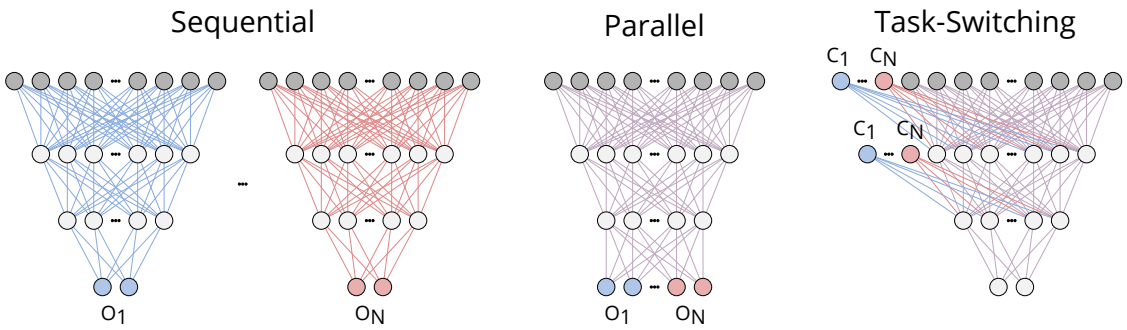

Figure 1: Architectures of multi-task learning. O_1 - O_N represent the outputs for each task. For the *task-switching* network, the task changes according to the binary context inputs C_A - C_N.

Results

We first analyzed the neural representations of three different multi-task learning networks (Figure 1). Similarly to Caruana (1997), we compared multitasking using *sequential* and *parallel* neural networks. In addition, we included *task-switching* networks, where tasks are informed by binary context inputs.

Task-switching **networks operate in a midway regime between** ***sequential*** **and** ***parallel*** **networks**

To analyze the neural representations learned by each network, we used the representational similarity analysis (RSA) (Kriegeskorte et al., 2008). We calculated the representational dissimilarity matrices (RDMs) for the different networks (Figure 2a) and proceeded by estimating the RSA (Figure 2b). The RDM was built by grouping, in terms of digit images, the neural activations of each layer. For each digit, we averaged the neural activity and computed their dissimilarity. We repeated this process for different tasks and calculated the dissimilarity between class labels (digits). The final RDM is a symmetric matrix where the main-diagonal blocks show the (dis-)similarity of activations within each task, and the off-diagonal blocks show the (dis-)similarity of activations between tasks. See methods section Representational Dissimilarity Matrix for further details on the calculation of the RDM.

The average RDM for the three models after training with *parity* and *value* for 10 different runs appear in Figure 2a. By observing the RDMs in layer 5, we can easily decode task-specific responses in *sequential* and *task-switching* networks, but not in *parallel* networks, for which the response of the two tasks is entangled. This is a pattern that we can also notice in layer 3 of *task-switching* networks, however after layer 3 the patterns evolve differently in the two types of networks.

To quantify the degree of similarity of representations between networks, we estimated the RSA by calculating the Pearson correlation between RDMs across runs. Results in Figure 2b show that *task-switching* networks behave similarly to the *parallel* networks in the first stage of processing, yet *task-switching* networks end up becoming more similar to the *sequential* networks. An extended analysis incorporating three additional tasks for the different models showed analogous results (Figure S7a and Figure S7b).

Parallel networks run the processing of all tasks simultaneously, which has been proposed to promote generalizability by regularizing the network through extracting shared representations between tasks (Caruana, 1997). Here, we formalized a metric for shared representation (degree of similarity of representations between tasks) from the RSA analysis and calculated its value in the different models across different layers. Shared representations were derived from the Pearson correlation of the off-diagonal blocks in the RDMs (see section Shared Representations). We calculated individually the shared representations for each model for each layer and then averaged the value across the 10 different runs. We represented the shared representations across layers for the different models in Figure 2c. In this graph, we observe that *parallel* and *task-switching* networks have a larger amount of shared representations between tasks than *sequential* networks, since task-specific *sequential* networks are disconnected from each other. *Task-switching* networks presented the highest amount of shared representations overall, which was further reinforced when we increased the number of tasks from two to five (Figure S7c).

To understand how representations were learned by the different models at different stages, we trained linear decoders to identify the features (digit, task and congruency information) encoded at different layers

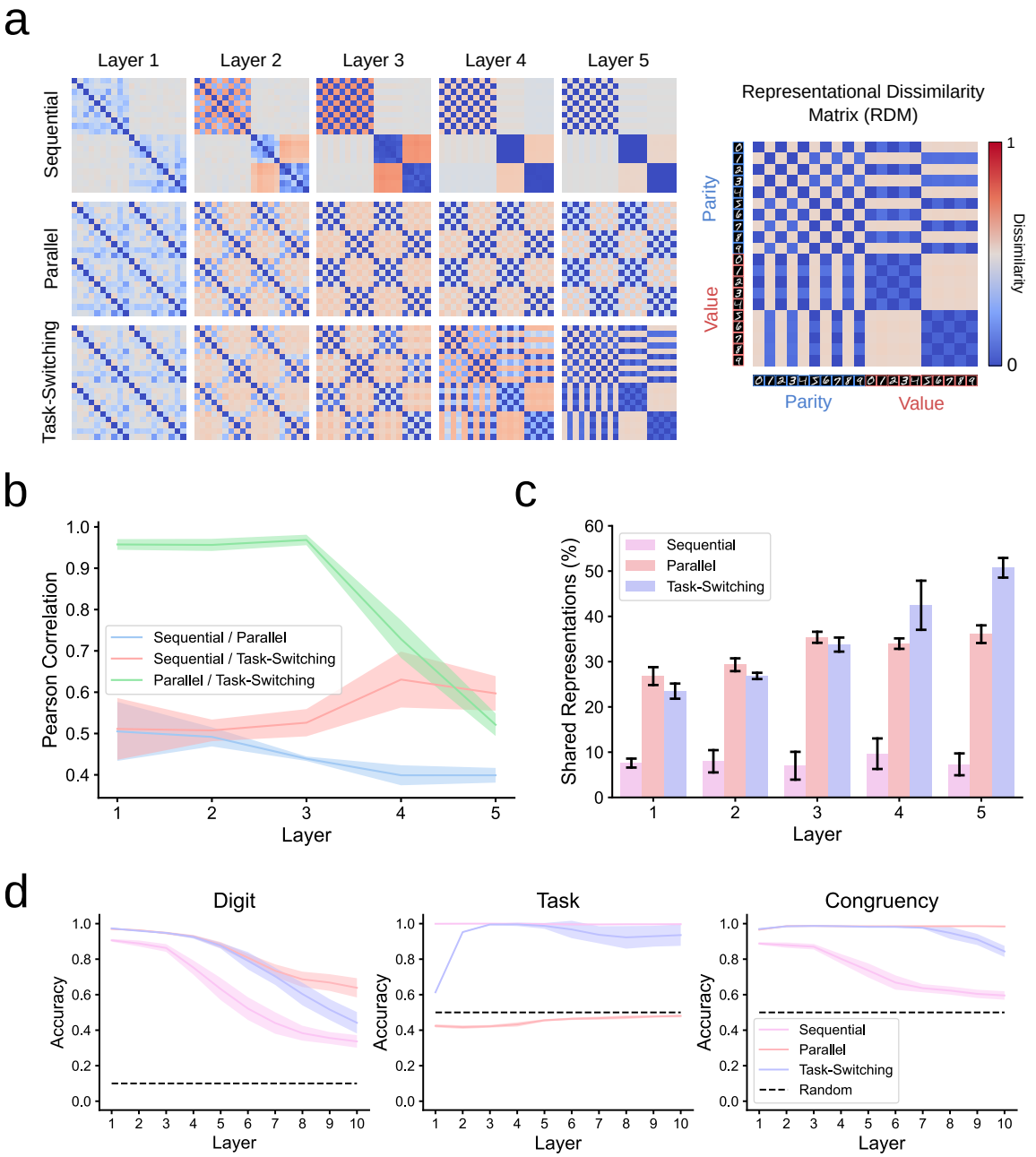

Figure 2: Neural Representations in Multi-Task Networks. (a) Average representational dissimilarity matrix (RDM) of the *parity/value task-switching* schedule for the three architectures across 10 different runs (Mean and SD). (b) Pearson correlation coefficient between RDMs across layers. (c) Shared representation of the models across layers. (d) Decoding accuracy for digit, task and congruency for models of 10 layers. Mean and SD across 10 different runs.

(Kriegeskorte & Douglas, 2019). For this analysis, we used models with up to 10 layers. The results from the linear decoders supported the view that *task-switching* networks lay in between *sequential* and *parallel* models (Figure 2d, all panels). Regarding the digit information, we observed that the information decreased as the number of layer increased for the three network models (Figure 2d, left). As for the task, we show that the decoder performed randomly when the activity of *parallel* networks was considered, indicative that task responses in *parallel* networks are entangled in intermediate layers (Figure 2d, middle). In contrast, task

responses were perfectly decoded from the activity of intermediate layers in *sequential* networks (Figure 2d, middle). The congruency information (Figure 2d, right) could be perfectly decoded from *parallel* networks, whereas there was a decrease in performance for *sequential* models, and also for *task-switching* models in latter processing stages. From this analysis we can infer that *parallel* networks resolve distinct tasks simultaneously by first extracting and keeping the congruency information through intermediate layers, so that specific task responses are materialized only in the output layer.

We assessed the performance of the three network architectures by analyzing the accuracy achieved across runs among tasks: Figure S8a shows a similar performance for the three networks in the two tasks, despite significant differences appeared between *sequential* and *parallel* networks, as well as between *sequential* and *task-switching* networks, when performance was evaluated simultaneously for the two tasks (Wilcoxon Signed-Rank Test, $p < 0.001$). This can be explained in terms of parameter sharing, i.e., weights and biases that are shared between tasks in *parallel* and *task-switching* networks, which regularize the network performance. We conducted the same analysis on the five different tasks obtaining similar results (Figure S8b).

Task-switching **networks perform multiple tasks by extracting task congruency information**

We then focused on *task-switching* networks to further analyze the behavior of their neural representations. In *task-switching* networks, tasks share a common pool of neural activations (i.e., shared representations) that are then biased based on context inputs towards resolving one task or another. To better understand this common computation, first we investigate the RSA by calculating the correlation between RDMs of three variants of the *task-switching* network with 10 layers: only context inputs in the first layer (*first*), context inputs in all layers (*all*), and a network trained with all context inputs, which were then removed at inference time (*removed*). Figure 3a shows the RSA between these networks: The *removed* and *all* networks have similar representations in the first layers, only to diverge as we approach the output layer. Contrarily, *first* and *all* start with lower correlation but then converge to the maximum correlation. The drop of correlation between these two networks in the first layers is explained by the early resolution of the task in the *first* network, which is caused by the presence of the context inputs only in the first layer. Note that context inputs are the only task-specific information in *task-switching* networks. Hence, the network is forced to separate the two tasks before the context input disappears.

Next, we computed multidimensional scaling (MDS) on the RDMs (Jazayeri & Ostojic, 2021) (for implementations details see methods section Multidimensional Scaling). Figure 3b shows the MDS for the first and last layers. In the *first* network, digits were represented differently for both tasks since the context inputs were introduced only in the first layer, as explained above. For the *all* and *removed* networks, we see that, by layer 3, the projection of digits is mostly orthogonal: one dimension for congruent stimuli, and the other for incongruent stimuli.

In addition to MDS, we used a generalized linear model to decode the activations in the original dimensional space (see methods section Generalized Linear Model). Figure 2d shows the performance of the classifier at decoding linearly for the specific digit, the task and the congruency. In high-dimensional space, digit is decoded with high accuracy in the first layers. For all three models, the accuracy collapses in the last layers. The *removed* and *all* networks are slightly better than the *first* network, due to the faster task processing of this network, as explained earlier. Consistently, task information is already present in the first layer of the *first* network, in clear contrast to the *all* network: the *all* network shows low accuracy in the first layers, as if the context inputs were not acting before the third layer (see also results section Context strength is maximized at intermediate layers). Finally, to test the hypothesis that *task-switching* networks process congruency information, we evaluated the performance of the linear decoder reconstructing the congruency. The *removed* network is capable of decoding the congruency perfectly across layers in the high-dimensional space defined by the neural activations, showing that the congruency processing is impaired at the time of resolving the task through context processing. This is consistent with the *first* network showing the fastest and greatest decrease in congruency decoding power as processing in the layers advances. Together, these plots confirm the hypothesis that context inputs bias neural activity towards the goal of the task. Congruency is the midpoint between the optimization of the two tasks and determines the amount of shared representations. In the extreme case of two identical tasks, the overlapping would be maximum and the bias signals would converge, reducing multitasking to a single task, for which context inputs would act as mere activation biases.

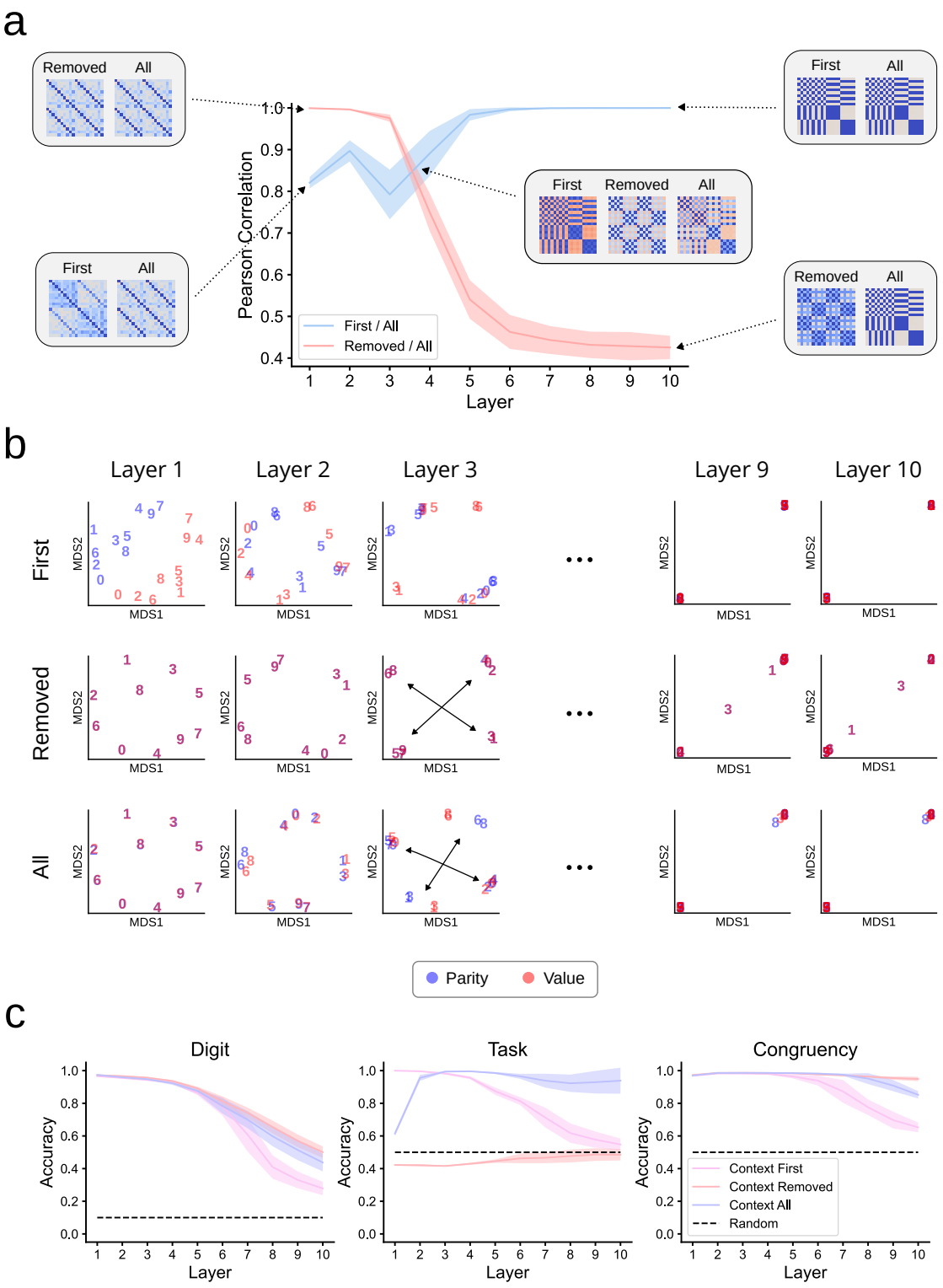

Figure 3: Layer specific neural representations: shared vs. dissimilar representations of Digit, Task and Congruency content. (a) Representational similarity analysis (RSA) between *first*, *all* and *removed* network for 10 different runs (Mean and SD). (b) Multidimensional scaling (MDS) of the representational dissimilarity matrix (RDM). (c) Linear decoder of neural activity for 10 different runs (Mean and SD).

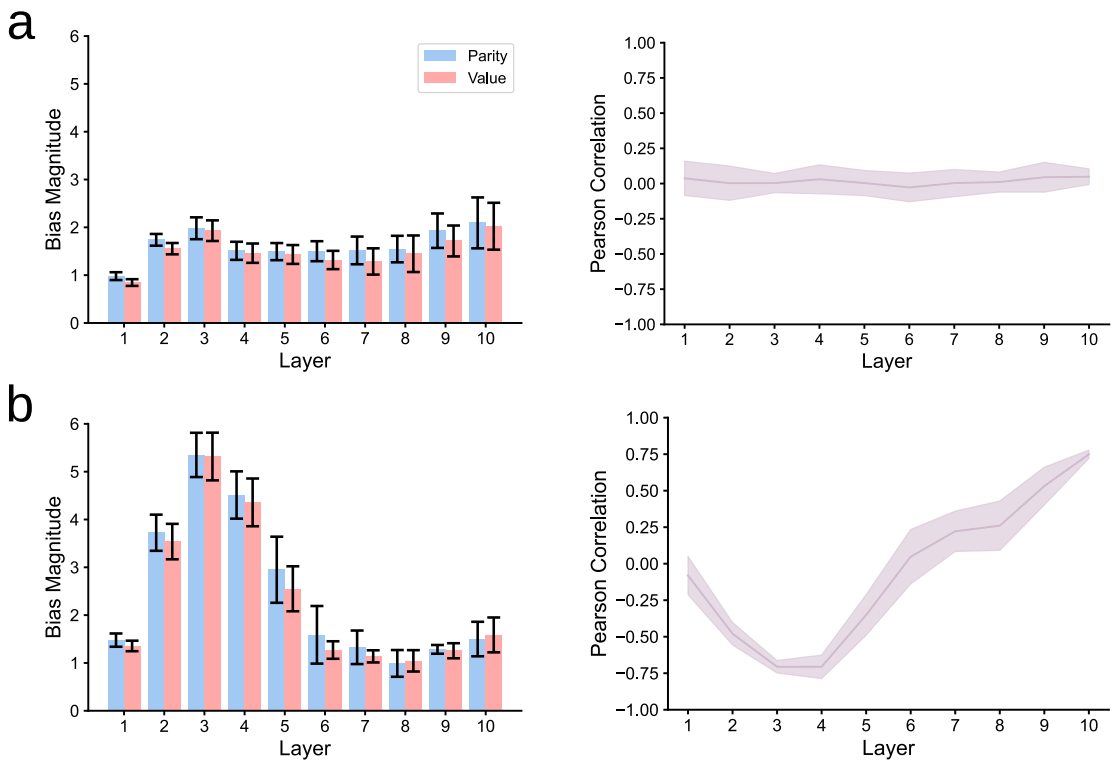

Figure 4: Weight magnitude and correlation between input biases in *sequential* networks (a), compared to *task-switching* networks (b). Note that input biases are associated with task contexts only in *task-switching* networks. The magnitude of the bias is calculated using the $L2$-norm (see methods section Bias Magnitude). Mean and SD shown over 10 runs.

Context strength is maximized at intermediate layers

After observing the impact of the context inputs in the first layer, we hypothesized that the location of the context information within the hierarchical neural processing organization could be important to optimize feature extraction aligned towards the task goal. Flesch, Juechems, et al. (2022) observed anticorrelation between the task biases induced by context inputs in networks with one hidden layer. Here, we run a series of model realizations to assess the strength and correlation of task biases when varying the position of the context inputs along the *task-switching* network. As control, we repeated the analysis in *sequential* networks, for which bias inputs mediate mere biases instead of task-related biases (Figure 4). Indeed, for *sequential* networks, the location of the biases was not sensitive neither in magnitude nor in terms of correlation. Only the first layer seemed to have a lower magnitude compared to others (Figure 4a, left). As for the correlation between biases from different *sequential* networks, the mean correlation fluctuated close to 0 (Figure 4a, right), indicating that representations in *sequential* networks are not shared across tasks (see Figure 2). In contrast, when analyzing the magnitude of *task-switching* networks trained for the same tasks, we observed that the magnitude increased considerably in the intermediate layers 2 to 5 (Figure 4b, left). The increase of magnitude in the intermediate layers suggests an important role of the context mediated biases to route task information. This was confirmed by the increase of anticorrelation between tasks in the same layers (Figure 4b, right).

In Figures S9, S10 and S11, we extended this analysis for *sequential* and *task-switching* networks with different numbers of context and layers. Each row in these figures represents a different context configuration and each column denotes the index of the layer (we fixed the maximum number of layers to 10). Colored cells according to the heatmap refer to layers with context inputs at that position (Figures S9, S10 and S11), whereas gray cells refer to layers with no context inputs (Figures S9 and S10), and missing cells in Figure S11 denote the absence of the layer, so these networks were shallower. We tried four different settings: adding context inputs starting from the first layer (Figure S9a), adding context inputs starting from the last layer (Figure S9b), adding a context inputs only at a particular layer (Figure S10), and adding layers with context inputs progressively

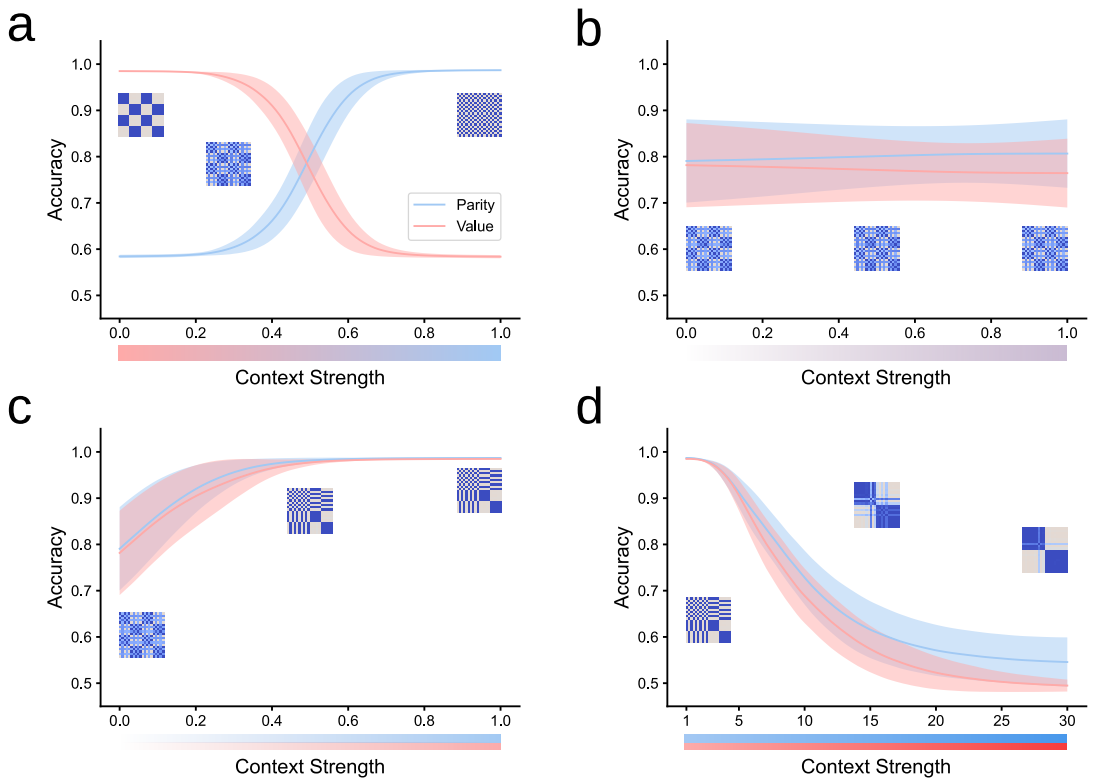

Figure 5: Task accuracy depends on context intensity (Mean and SD for 10 different runs). (a) The total context strength is held fixed to 1. (b) Increased context strength in both tasks at once. (c, d) Increased context strength in each task independently.

(Figure S11). For all these conditions, we calculated the mean magnitude for each task (left and center panels) and the correlation between context-mediated task biases (right panels).

The results show that the largest bias strength and task anticorrelation was achieved at intermediate layers centered around layers 3 and 4 (Figures S9 and S11). Yet, imposing the context inputs either only early in the processing, or only at the end was also associated with a large bias magnitude and strong anticorrelation (Figures S9a and b), which is explained by the fact that the network is forced to process digit information and task separation simultaneously in the case of early processing of context inputs, and task separation and the final response in the case of late processing of context inputs. However, when context inputs were present layer after layer, positive correlations also emerged between the two tasks in the last processing stages (Figures 4b right panel, S9 and S11).

Task-switching networks generalize representations between tasks

Next, we analyzed the robustness of *task-switching* networks to perturbations added to the learned task biases. First, we interpolated the values of contexts inputs between [0,1] for the two tasks keeping their sum to 1 (see methods section Bias Interpolation). A bias interpolation in this way can be related to uncertainty, either due to task inference in uncued paradigms (i.e., internal estimation of the likelihood to perform a task or another), or in cued, but stochastic, task designs. Figure 5 shows the accuracy for the complementary values of the two context inputs. We moved from task *value* to task *parity*. Results show that the transition between tasks was smooth: there was not a hard threshold at 0.5, but a progressive change from one task to the other. The neural representations at 0.5 showed an intermediate state between tasks. The baseline accuracy was 0.6 since it was the level of congruency between tasks. The accuracy at 0.5 was close to 0.8, which was the chance level once the congruency baseline was taken into account.

We then incremented the value of both contexts together from 0 to 1. This is a way to parametrically analize performance according to context strength when both tasks are equally probable. Figure 5b shows the accuracy

values for both tasks. The accuracy began at 0.8 and slightly diverged the higher the context strength, which proved a slight preference of the trained networks for the *parity* task. The internal representations however, showed that the network operated midway between the two tasks, generalizing the observation in Figure 5a when both tasks had the context strength equal to 0.5.

We also analyzed the case when one context strength was varied keeping the other at zero, and did so for values lower and greater than one. Lower than one values can be related to non-normalized task uncertainties, whereas greater than one values are useful to infer behavior against aberrant context inputs. In Figure 5c, we swept each task independently from 0 to 1, and represented the values in the same plot. Results indicate that networks did not need to be fully certain of the magnitude of the context input to operate at highest performance. Concretely, we observed that with contexts close to 0.5, the networks reached the maximum accuracy. From each network we computed the RDM. The results showed that the neural representations for binary (full certain) contexts were already present for uncertain contexts as well, if their strengths were about 0.5 or greater.

Analyzing the behavior of the *task-switching* network against abnormal context values, greater than one, was interesting to account the task performance under aberrant context modulations, such as those that could be originated from abnormal activity in the prefrontal cortex. Figure 5d shows the accuracy for each task when the values of their contexts increased up to 30-fold their values during training. When the context dramatically increased, the network was unable to process the stimulus (digit) upon which determine the proper task response, which led to a decrease in accuracy: the information about the task was enhanced at the cost of masking the information about the digit. Hence, the dissimilarity between mean activations in the output layer became close to zero.

Mixed selectivity emerges for resolving task execution

After having inspected neural representations through layers, we investigated the role of single neurons, especially in the last layer, where the network needs to predict the proper output response. We hypothesized that neurons have to be organized to drive their activity collectively towards the task goal. Mixed selectivity was previously observed in neural networks with a single hidden layer and context inputs in the first layer (Flesch, Juechems, et al., 2022). Here, we analyzed the activations of neurons in the last layer through agglomerative hierarchical clustering (Figure 6, further technical details in the methods section Agglomerative Hierarchical Clustering).

The results of clustering are shown in Figure 6. The different plots represent the mean activity of the different clusters for each task when presented with one of the digits. We used the *all* and *removed* networks from the previous section. The activity was normalized to 1 (radius of the plot).

We observed the existence of two different clustering patterns across the 10 simulations. A first pattern of activation is showed in the first row of Figure 6. We found two clusters of activity, each representing a different class. For the parity task, we found a cluster for even digits (in blue) and another cluster for odd digits (in red). And similarly for the value task, regarding values smaller than or equal to four (in blue), vs. larger (in red). The congruent digits showed the highest responses. In fact, when context inputs were removed, the clusters only encoded the congruent digits, entailing that context inputs are responsible for the increase in activity of incongruent stimuli. In addition to the congruent/incongruent asymmetry, this pattern of activity also showed an asymmetry between classes: in the parity task, the responses to even digits were dominant against odd digits, and similarly for the responses to values smaller than or equal to four against larger than four in the Value task.

The second row of the plot represents the second pattern. As in the first pattern, the congruent digits showed the highest responses. In contrast to the first pattern, however, the responses for the non-dominant classes were apparently missing. This reflects a strong magnitude dissimilarity between dominant and non-dominant classes in this case. In fact, pattern 2 represents an extreme case of pattern 1 when the ratio of non-dominant vs. dominant responses tends to zero. We observed that from 10 runs, 7 converged to pattern number 1, suggesting that learning favored neurons to be tuned to different tasks depending on the current active stimulus and context, resembling mixed selectivity in the brain (Rigotti et al., 2013), i.e., neurons that are responsive to multiple task dimensions.

We repeated the analysis using five tasks and found pattern 1 (Figure S12a) and pattern 2 (Figure S12b) in the same proportion. This is indicative that learning similarly favors a default network mode, even when the number of tasks increases.

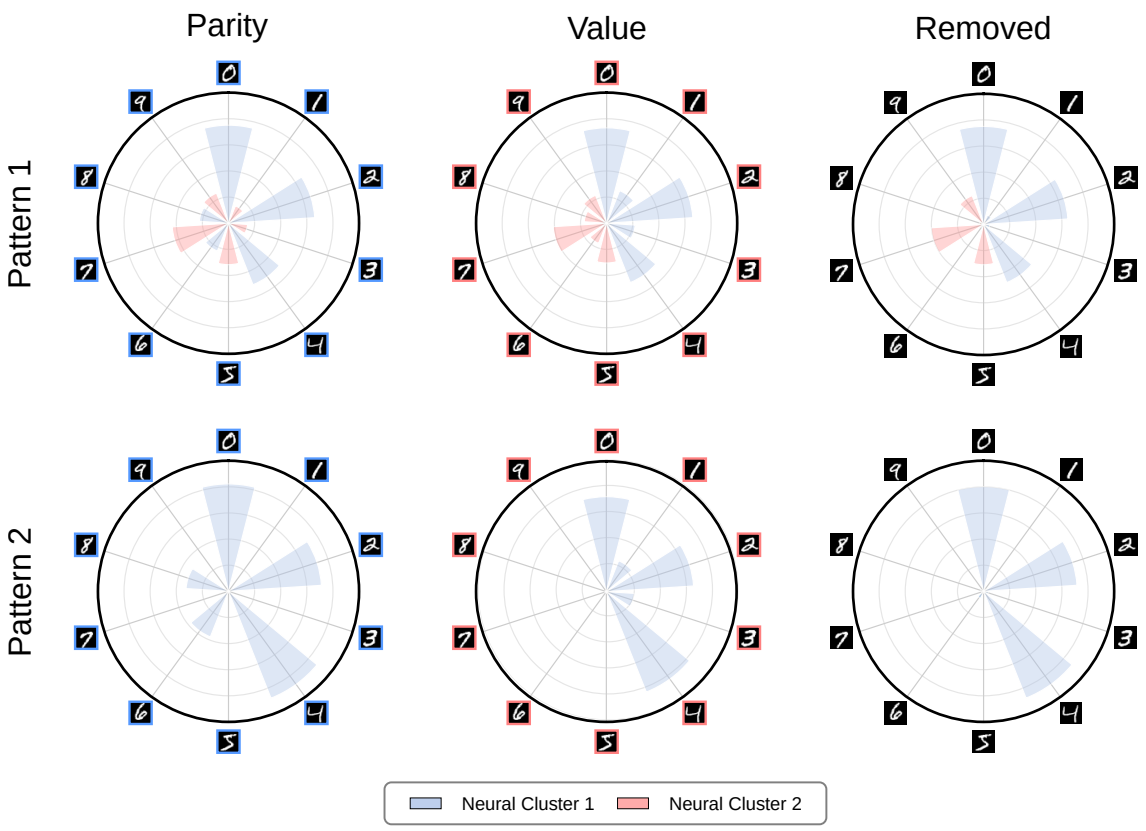

Figure 6: Neural clustering for the *parity/value task-switching* schedule.

All together, these patterns help explaining two different ways by which the network derives the proper output to each stimulus conditions. There is a default mode that emerges through learning, in which the output bias favors a specific output response, i.e., the response for a particular class. The patterns of activity presented here in the last layer are therefore responsible to overcome the output bias when appropriate, explaining the strong response asymmetry between classes. In that sense, the default mode of the network is in fact closely related to the non-dominant responses. Our analysis emphasizes that the dominance at short scale (e.g., in a subset of layers) might be misleading with respect to the dominance throughout network computation.

Discussion

In this study, we conducted a series of analyses with multi-task networks, namely *sequential*, *parallel* and *task-switching* networks. We studied these networks using representational similarity analysis (Kriegeskorte et al., 2008; Kriegeskorte & Kievit, 2013), a methodology developed in neuroscience to relate task stimuli with internal representations, in our case, neural activations in each layer of the neural network. We focused mainly on two tasks, parity and value, which have been used in human experiments (Monsell, 2003) and in artificial neural networks (Bernardi et al., 2020). We utilized a definition of shared representations between tasks based on representational dissimilarity matrices, which is computed from the off-diagonal blocks of the correlation matrix. This analysis indicates that *task-switching* networks possess a significantly larger proportion of shared representations compared to *parallel* networks in the last layers of processing (Figure 2c). As we increased the number of tasks, however, shared representations between tasks stopped increasing with layer processing (Figure S7c). Only *task-switching* networks showed a remarkable increase of shared representations in the last layer.

Our study shows that *task-switching* networks learn disentangled shared representations that allow to reconstruct the mapping between task and goal. We trained linear decoders to study how layers extracted features along the processing in the network (Kriegeskorte & Douglas, 2019). Context inputs can either improve feature

extraction or interfere with the network processing, depending on their position in the network. When all contexts inputs are removed, *task-switching* networks function as congruency detectors to resolve multitasking (Figure 3c), similar to *parallel* networks (Figure 2d). On the other hand, training *task-switching* networks with the context input only in the first layer precludes the extraction of common task features, similar to what happens in disconnected *sequential* networks. These results suggest that *sequential* and *parallel* networks can be seen as particular *task-switching* networks, where the context inputs are diminished (*parallel* networks), or introduced in the early layers of processing (*sequential* networks). This picture also emerges when contexts are continuously modified between [0,1] (Figure 5c; the pattern of *parallel* shared representations appear for the lowest context inputs) and beyond 1 (Figure 5d; the pattern of absent shared representations characteristic of *sequential* networks appear for the largest context inputs).

We estimated the importance of task biases in the network processing by calculating the norm for the different layers. We understand biases with higher norm to have more importance at driving network activity towards resolving one task. This analysis has been previously done for shallow neural networks with one layer (Flesch, Juechems, et al., 2022). Here we generalized the results for neural networks of up to 10 layers, with contexts placed at different locations. Results show that the third and fourth layers have greater importance at modulating downstream activation (Figure 4b, Figure S9 top panel and Figure S11). Moreover, the higher anticorrelation between task contexts coincide in these layers (Figure 4b, Figure S9 top panel and Figure S11). The first and second layer play a larger role whenever the neural network is shallow (Figure S11). However, we also see that contexts that are only located in the last layer have the greatest absolute magnitude (Figure S9, bottom panel). Additionally, if only a single context input is added, the resulting task biases become highly anticorrelated (Figure S10).

Finally, we wanted to observe the role of single neurons, which can be done by studying task selectivity in artificial neural networks (Leavitt & Morcos, 2020; Flesch, Juechems, et al., 2022). We observed that neurons in the last layer cluster together for different tasks according to the output of the network. These results support the idea that contexts drive the activation of different subnetworks, promoting the selectivity of single neurons to several tasks.

We have seen that context inputs help the encoding of tasks. We tested this in simple networks performing binary multitasking upon the MNIST dataset. In more complex architectures, such as convolutional neural networks, the usage of context inputs has been also explored (Li et al., 2016; Liu et al., 2019; Sun et al., 2021), but a characterization of the neural representations is still pending. Further research following the lines of the present study should focus on convolutional neural networks and other sophisticated architectures, where context inputs have also been employed, such as in alleviating catastrophic forgetting in continual learning (Masse et al., 2018; Serra et al., 2018; Grewal et al., 2021; Flesch, Nagy, et al., 2022), where the study of neural representations across time could help understanding the underlying mechanisms. This represents a first step towards developing more explainable and robust mechanisms.

Methods

Neural Network Architectures

We conducted different analyses with three variants of the feedforward network (Rumelhart et al., 1986; Goodfellow et al., 2016) in the context of multi-task learning.

Sequential Networks: Each task is learned by an independent network. multitasking is performed by combining the outputs of the networks. A *sequential* network is parameterized as $\mathbf{y}_t = f_t(\mathbf{x}; \mathbf{W}_t, \mathbf{b}_t)$, where $t \in T$ denotes the task in the set $T = \{T_1, T_2, ..., T_N\}$ of N tasks, and $\mathbf{W}_t$ and $\mathbf{b}_t$ denote respectively the synaptic weights and biases specific for each task in each layer. These parameters are independent, hence not shared across tasks.

Parallel Networks: Tasks are processed simultaneously and the output layer accounts for independent responses for each of the tasks. The *parallel* processing regularizes the network and allows overall better performance than the *sequential* networks (Caruana, 1997). A *parallel* network is defined as $\mathbf{y}_T = f(\mathbf{x}; \mathbf{W}, \mathbf{b})$. The parameters in hidden layers are exactly the same for both tasks, hence fully shared. The only independent parameters in this network are the ones present in the output layer.

Task-switching Networks: Each task is processed individually, but parameters, other than the task biases, are fully shared. The output of a *task-switching* network is $\mathbf{y}_t = f_t(\mathbf{x}; \mathbf{W}, \mathbf{b}_t)$. Weights are co-optimized for all tasks, but the task bias vectors $\mathbf{b}_t$ are different. We define a *task-switching* network to be a composition of layers, which can incorporate the context input for each task, $\mathbf{z} = \mathbf{W^T a} + \mathbf{b}_t$ or not $\mathbf{z} = \mathbf{W^T a}$, where

$\mathbf{a}$ denotes the activations of the previous layer and $\mathbf{z}$, the output from the current layer. A *task-switching* network requires at least one layer with the context input. We train *task-switching* networks following the interleaved *task-switching* schedule (Flesch et al., 2018). The three different architectures analyzed in this study are depicted in Figure 1.

Binary Tasks

In this study, we created five *binary tasks* using the MNIST dataset (LeCun et al., 1998). The main task pair we use throughout the study corresponds to *parity* and *value*, which have been used in human experiments (Monsell, 2003) and in artificial neural networks (Bernardi et al., 2020). Parity defines a digit as either even or odd, whereas value defines a digit being larger than four, or not. In total, we designed up to five tasks, all using the MNIST dataset. For N tasks, we can also calculate the joint accuracy of tasks using the chain rule of probability:

$$\mathbb{P}(T_1 \cap T_2 \cap \cdots \cap T_N) = \prod_{i=1}^{N} \mathbb{P}(T_i),$$

We analyzed the impact of *congruency* in learning representations. For two different tasks, a stimulus (digit) is *congruent* if the proper response is encoded by the same output neuron. The opposite, that different output neurons encode the proper responses for the two tasks, is referred as *incongruency*.

Table 1 enumerates the tasks and the expected output of the network, i.e., the labels (0, for the first output unit, and 1 for the second). The congruency between tasks is represented in Table 2.

Task Name	Number									
	0	1	2	3	4	5	6	7	8	9
Parity	1	0	1	0	1	0	1	0	1	0
Value	1	1	1	1	1	0	0	0	0	0
Prime	0	0	1	1	0	1	0	1	0	0
Fibonacci	1	1	1	1	0	1	0	0	1	0
Multiple of 3	0	0	0	1	0	0	1	0	0	1

Table 1: Task names and output labels.

	Parity	Value	Prime	Fibonacci	Multiple of 3
Parity	1	0.6	0.3	0.5	0.4
Value	0.6	1	0.5	0.7	0.4
Prime	0.3	0.5	1	0.6	0.5
Fibonacci	0.5	0.7	0.6	1	0.3
Multiple of 3	0.4	0.4	0.5	0.3	1

Table 2: Congruency ratio among tasks.

Training

For each architecture, we trained two different models with 5 (results section *Task-switching* networks operate in a midway regime between *sequential* and *parallel* networks) and 10 layers (results sections *Task-switching* networks perform multiple tasks by extracting task congruency information, Mixed selectivity emerges for resolving task execution, Context strength is maximized at intermediate layers, *Task-switching* networks generalize representations between tasks). Networks were trained for a total number of 50 epochs. We split the images into a training set of 50,000 images and a test set of 10,000. The batch size was fixed to 100 images. The number of hidden neurons per layer was fixed to 100. We selected ReLU as the activation function (Glorot et al., 2011) and initialized weights and biases following a uniform distribution in the range $(-1/\sqrt{N}, 1/\sqrt{N})$, where N is the number of inputs to the layer. We used the Adam optimizer for training (Kingma & Ba, 2014)

with learning rate 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and the cross entropy loss function. The models were implemented in PyTorch (Paszke et al., 2019). We collected statistics over 10 different runs and controlled the random number generator for reproducibility and hypothesis testing. We use the Wilcoxon Signed-Rank Test to test statistical significance between models (Demšar, 2006).

Representational Dissimilarity Matrix

For each layer, we construct an activation matrix of dimension $N_{test} \times N_{hidden}$ for the N tasks, where N_{test} is the number of digits in test set and N_{hidden} is the number of hidden units in the layer. For each digit 0–9, we calculate the average activation for the hidden units in a layer. For each task, we then assemble a matrix of dimension $N_D \times N_{hidden}$, where N_D is the number of digits. This matrix represents the average unit response to each of the possible stimulus in the dataset. We concatenate the average activations for all tasks and calculate the cross-correlation to create the correlation matrix $\mathbf{R}$, where $\mathbf{R_{ij}}$ denotes the pairwise Pearson correlation between average activation vectors $\mathbf{a_i}$ and $\mathbf{a_j}$. The normalized representational dissimilarity matrix, RDM is calculated by scaling $\mathbf{R}$, where:

$$\text{RDM} = \frac{1 - \mathbf{R}}{2},$$

where RDM is bounded in the range $[0, 1]$, where 0 represents similarity (correlation) and 1 complete dissimilarity (anticorrelation).

For *sequential* and *task-switching* networks, calculating the dissimilarity between tasks can be done extracting directly the mean activation for each layer and task. For *sequential* networks, we have specific activity patterns from each network performing each task, and for *task-switching* networks, we have specific activity patterns originated from the same network under each task context. For *parallel* networks, since the internal processing for both tasks is exactly the same, the resulting neural representations are entangled. Therefore, the RDM is composed of tile repetitions of the same activation patterns.

Shared Representations

To calculate the shared representations between tasks, we calculate first the RDM, which is formed by submatrices or blocks of $N_D \times N_D$ that conform the main diagonal and the off-diagonal, where N_D is the number of digits. The total number of blocks of the RDM, N_B, is $N_{MD} + 2N_{OD}$, where N_{MD} is the sum of the blocks in the main diagonal and N_{OD} the blocks of the off-diagonal. We interpret shared representations (SR) to be correlated vectors in the off-diagonal blocks.

$$\text{SR} = \frac{1}{N_{OD} N_D^2} \sum_k^{N_{OD}} \sum_i^{N_D} \sum_j^{N_D} |\mathbf{R}_{ij}^k|,$$

where N_{OD} is given by ${}^{N_T(N_T-1)}/_2$, where N_T is the number of tasks, and $\mathbf{R}_{ij}^k$ corresponds to the correlation at position ij of block k.

In addition, we can calculate the task representations (TR) by calculating the overall correlations in the main diagonal block:

$$\text{TR} = \frac{1}{N_{MD} N_D^2} \sum_k^{N_{MD}} \sum_i^{N_D} \sum_j^{N_D} |\mathbf{R}_{ij}^k|,$$

N_{MD} is the number of blocks in the main diagonal, which corresponds to the number of tasks, N_T. We could calculate the complete representations (CR) of the RDM as:

$$\text{CR} = \text{TR} + 2\text{SR},$$

since $N_B = N_{MD} + 2N_{OD}$ and the RDM is a symmetric.

Generalized Linear Model

We used the multiclass logistic regression as linear decoder of neural activations across layers. We used the scikit-learn module to implement the logistic regression model (Pedregosa et al., 2011). The maximum number

of iterations was set to 5000 to guarantee convergence with a tolerance of 10^{-3}. We used L2 regularization and cross entropy as loss function.

We used the activation matrix described in the methods section Representational Dissimilarity Matrix. Then, we divided the rows into a train set and test set following a 90/10 split and used digit labels to create the task and congruency labels. Finally, we assembled the activation matrix for each layer of a network and averaged the results across the 10 different runs.

Multidimensional Scaling

To construct the MDS embedding, we create first the RDM of a layer. RDM is a distance matrix where the position ij denotes the distance between i and j. We project this high-dimensional points into a two-dimensional embedding using MDS. We used the scikit-learn module to implement MDS (Pedregosa et al., 2011). The maximum number of iterations was set to 1000 to guarantee convergence with a tolerance of 10^{-5}.

Bias Magnitude

The magnitude of the task bias in a layer, $\mathbf{b}$, is calculated using the $L2$-norm:

$$||\mathbf{b}|| = \sqrt{\langle \mathbf{b}, \mathbf{b} \rangle} = \sqrt{\sum_{i=1}^{N} \mathbf{b_i^2}},$$

where $\mathbf{b_i}$ represents the task bias to neuron i; i running all neurons in the layer from 1 to N.

Bias Interpolation

Given two task biases $\mathbf{b_1}$ and $\mathbf{b_2}$, we can find a bias $\bar{\mathbf{b}}$ which is a linear combination $\mathbf{b_1}$ and $\mathbf{b_2}$:

$$\bar{\mathbf{b}} = \lambda_1 \mathbf{b_1} + \lambda_2 \mathbf{b_2},$$

By adding the constraint $\lambda_1 + \lambda_2 = 1$, we can interpolate smoothly from $\mathbf{b_1}$ to $\mathbf{b_2}$ by sweeping λ_1:

$$\bar{\mathbf{b}} = \lambda_1 \mathbf{b_1} + (1 - \lambda_1) \mathbf{b_2},$$

or from $\mathbf{b_2}$ to $\mathbf{b_1}$ by sweeping λ_2:

$$\bar{\mathbf{b}} = (1 - \lambda_2) \mathbf{b_1} + \lambda_1 \mathbf{b_2},$$

Agglomerative Hierarchical Clustering

Similarly as in the assembling of the RDM, we average the neural activity for the last layer for each digit, creating a matrix of size $N_D \times N_{hidden}$, where N_D is the number of digits and N_{hidden} is the number of units in the hidden layer. We gather the averages of activity for the different tasks to construct a matrix of size $N_D N \times N_{hidden}$, where N are the number of tasks. Each independent column conforms the average of activations across tasks and digits for a single hidden unit. We normalize the activity by the maximum value so the new maximum is set to 1. Upon these activations, we apply hierarchical clustering to create the dendrogram of hidden units using the *centroid* method in SciPy (Virtanen et al., 2020). We then find the cluster of each hidden unit and, finally, we calculate the average activity of each cluster per task and per digit.

Code

The code for reproducing the experiments is available at www.github.com/LofNaDI/neural-representations-multitasking.

Acknowledgements

This work was supported by the Generalitat Valenciana Gen-T Program (Ref. CIDEGENT/2019/043), Grant Ref. PID2020-120037GA-I00 funded by MCIN/AEI/10.13039/501100011, and European Union NextGenerationEU/PRTR - Programa de Planes Complementarios I+D+i (Ref. ASFAE/2022/014).

References

Ardid, S., & Wang, X.-J. (2013). A tweaking principle for executive control: neuronal circuit mechanism for rule-based task switching and conflict resolution. *Journal of Neuroscience*, *33*(50), 19504–19517.

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

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological review*, *108*(3), 624.

Caruana, R. (1997). Multitask learning. *Machine learning*, *28*(1), 41–75.

Cohen, J. D., Dunbar, K., & McClelland, J. L. (1990). On the control of automatic processes: a parallel distributed processing account of the stroop effect. *Psychological review*, *97*(3), 332.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, *7*, 1–30.

DiCarlo, J. J., Zoccolan, D., & Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, *73*(3), 415–434.

Flesch, T., Balaguer, J., Dekker, R., Nili, H., & Summerfield, C. (2018). Comparing continual task learning in minds and machines. *Proceedings of the National Academy of Sciences*, *115*(44), E10313–E10322.

Flesch, T., Juechems, K., Dumbalska, T., Saxe, A., & Summerfield, C. (2022). Orthogonal representations for robust context-dependent task performance in brains and neural networks. *Neuron*, *110*(7), 1258–1270.

Flesch, T., Nagy, D. G., Saxe, A., & Summerfield, C. (2022). Modelling continual learning in humans with hebbian context gating and exponentially decaying task signals. *arXiv preprint arXiv:2203.11560*.

Fusi, S., Miller, E. K., & Rigotti, M. (2016). Why neurons mix: high dimensionality for higher cognition. *Current opinion in neurobiology*, *37*, 66–74.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315–323).

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT Press.

Grewal, K., Forest, J., Cohen, B. P., & Ahmad, S. (2021). Going beyond the point neuron: Active dendrites and sparse representations for continual learning. *bioRxiv*.

Ito, T., Klinger, T., Schultz, D., Murray, J., Cole, M., & Rigotti, M. (2022). Compositional generalization through abstract representations in human and artificial neural networks. *Advances in Neural Information Processing Systems*, *35*, 32225–32239.

Ito, T., & Murray, J. D. (2021). Multi-task representations in human cortex transform along a sensory-to-motor hierarchy. *bioRxiv*.

Jazayeri, M., & Ostojic, S. (2021). Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Current opinion in neurobiology*, *70*, 113–120.

Johnston, K., Levin, H. M., Koval, M. J., & Everling, S. (2007). Top-down control-signal dynamics in anterior cingulate and prefrontal cortex neurons following task switching. *Neuron*, *53*(3), 453–462.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kriegeskorte, N., & Douglas, P. K. (2019). Interpreting encoding and decoding models. *Current opinion in neurobiology*, *55*, 167–179.

Kriegeskorte, N., & Kievit, R. A. (2013). Representational geometry: integrating cognition, computation, and the brain. *Trends in cognitive sciences*, *17*(8), 401–412.

Kriegeskorte, N., Mur, M., & Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 4.

Leavitt, M. L., & Morcos, A. (2020). Selectivity considered harmful: evaluating the causal impact of class selectivity in dnns. *arXiv preprint arXiv:2003.01262*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, H., Ouyang, W., & Wang, X. (2016). Multi-bias non-linear activation in deep neural networks. In *International conference on machine learning* (pp. 221–229).

Liu, S., Johns, E., & Davison, A. J. (2019). End-to-end multi-task learning with attention. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 1871–1880).

Mante, V., Sussillo, D., Shenoy, K. V., & Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, *503*(7474), 78–84.

Masse, N. Y., Grant, G. D., & Freedman, D. J. (2018). Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *Proceedings of the National Academy of Sciences*, *115*(44), E10467–E10475.

Menon, V., & D'Esposito, M. (2022). The role of pfc networks in cognitive control and executive function. *Neuropsychopharmacology*, *47*(1), 90–103.

Miller, E. K., & Cohen, J. D. e. a. (2001). An integrative theory of prefrontal cortex function. *Annual review of neuroscience*, *24*(1), 167–202.

Monsell, S. (2003). Task switching. *Trends in cognitive sciences*, *7*(3), 134–140.

Musslick, S., Saxe, A., Özcimder, K., Dey, B., Henselman, G., & Cohen, J. D. (2017). Multitasking capability versus learning efficiency in neural network architectures..

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … others (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Ravi, S., Musslick, S., Hamin, M., Willke, T. L., & Cohen, J. D. (2020). Navigating the trade-off between multi-task learning and learning to multitask in deep neural networks. *arXiv preprint arXiv:2007.10527*.

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., … others (2019). A deep learning framework for neuroscience. *Nature neuroscience*, *22*(11), 1761–1770.

Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K., & Fusi, S. (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature*, *497*(7451), 585–590.

Rougier, N. P., & O'Reilly, R. C. (2002). Learning representations in a gated prefrontal cortex model of dynamic task switching. *Cognitive Science*, *26*(4), 503–520.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Saxena, S., & Cunningham, J. P. (2019). Towards the neural population doctrine. *Current opinion in neurobiology*, *55*, 103–111.

Serra, J., Suris, D., Miron, M., & Karatzoglou, A. (2018). Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning* (pp. 4548–4557).

Sun, G., Probst, T., Paudel, D. P., Popović, N., Kanakis, M., Patel, J., … Van Gool, L. (2021). Task switching network for multi-task learning. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 8291–8300).

Vilarroya, O. (2017). Neural representation. a survey-based analysis of the notion. *Frontiers in psychology*, *8*, 1458.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … others (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, *17*(3), 261–272.

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., & Wang, X.-J. (2019). Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, *22*(2), 297–306.

Yuste, R. (2015). From the neuron doctrine to neural networks. *Nature reviews neuroscience*, *16*(8), 487–497.

SI Supplementary Figures

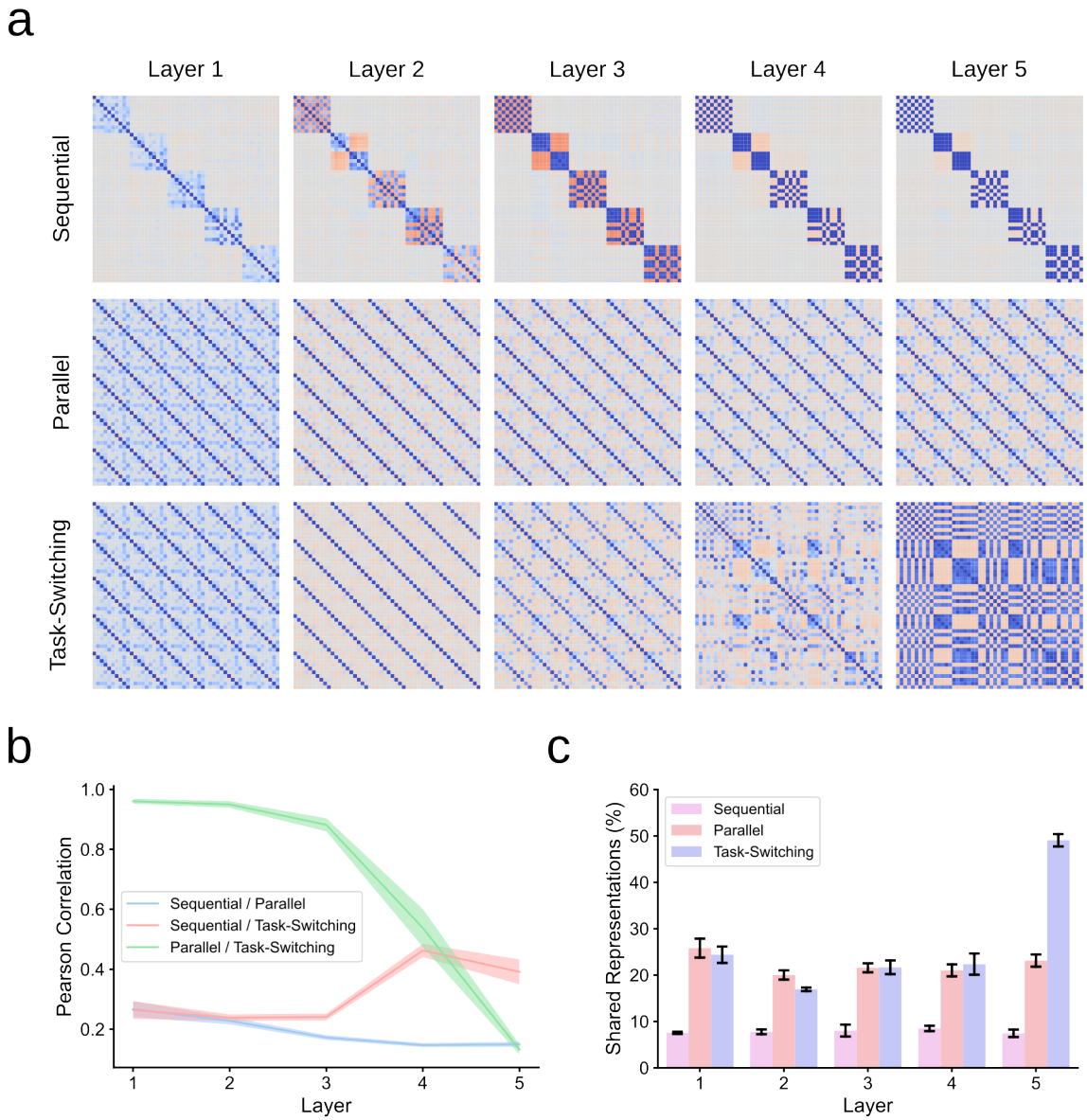

Figure S7: Neural Representations in Multi-Task Networks (analysis for the five tasks). (a) Average representational dissimilarity matrix (RDM) of the {*parity, value, prime, fibonacci, multiple of 3*} *task-switching* schedule for the three architectures across 10 different runs. (b) Pearson correlation coefficient between RDM (Mean and SD). (c) Accuracy of the models.

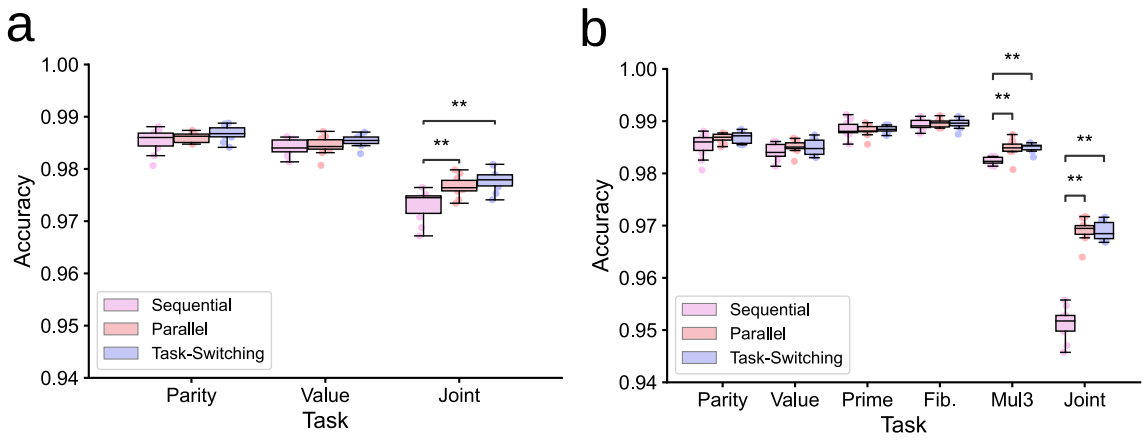

Figure S8: Accuracy for five-layer *sequential*, *parallel* and *task-switching* networks (Wilcoxon Signed-Rank Test, $p < 0.001$) for {*parity, value*} (a), and for {*parity, value, fibonacci, prime, multiple of 3*} (b).

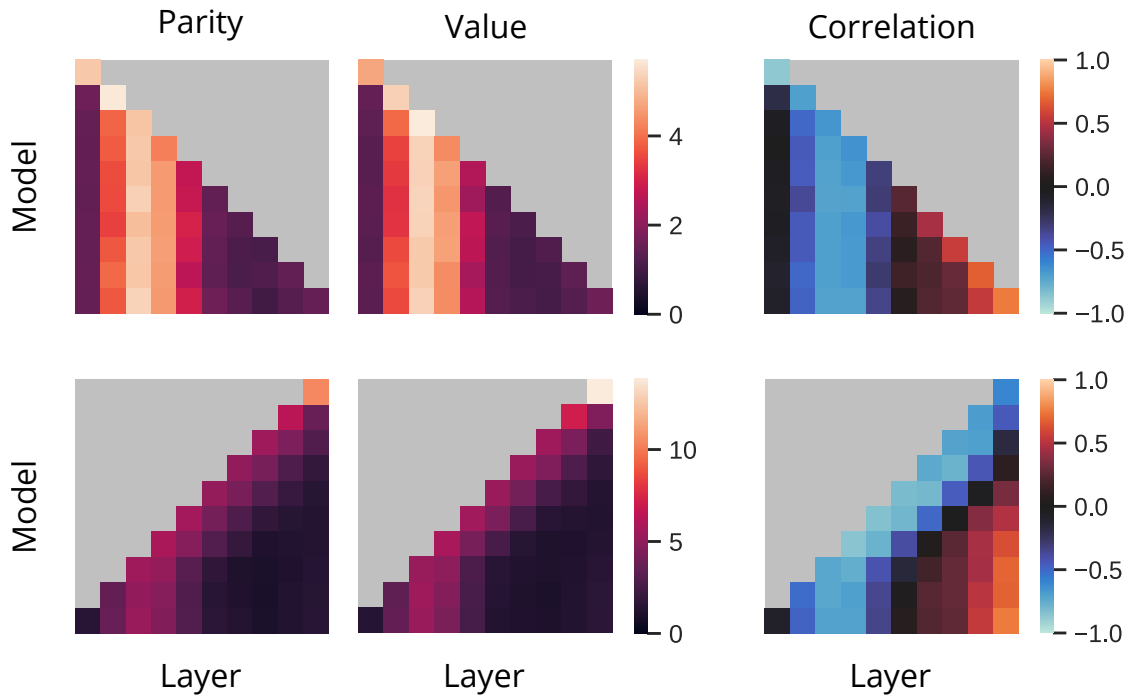

Figure S9: Mean magnitude and correlation between contexts added from layer 1 to layer 10 (panel a), or from layer 10 to layer 1 (panel b), using network architectures of 10 layers for 10 different runs. The gray background denotes the presence of a layer without context.

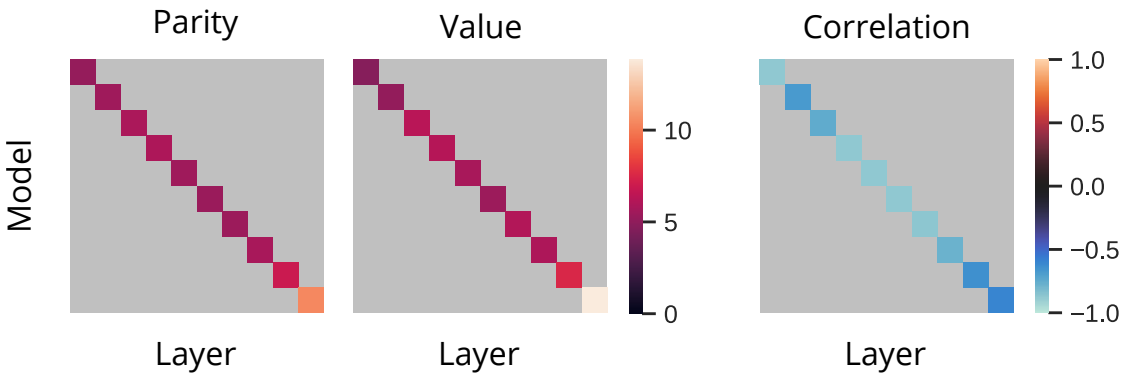

Figure S10: Mean magnitude and correlation between contexts (only a single task context is used, which is moved from layer 1 to layer 10), using network architectures of 10 layers for 10 different runs. The gray background denotes the presence of a layer without context.

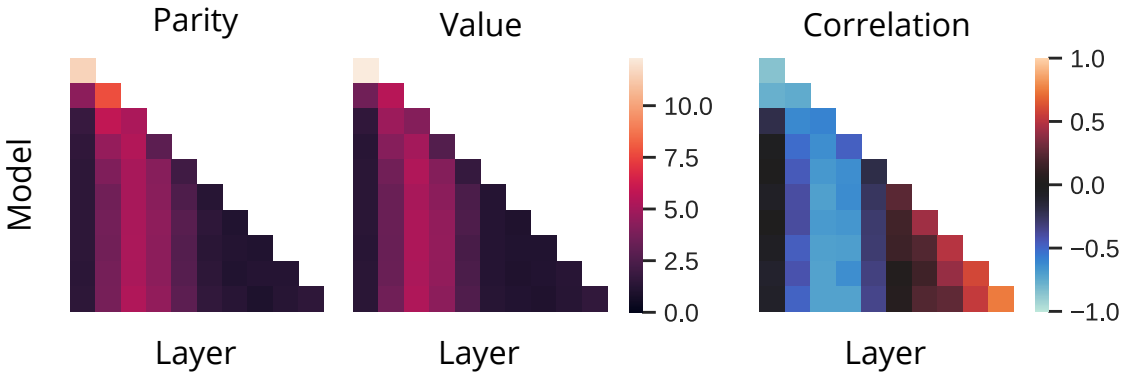

Figure S11: Mean magnitude and correlation between contexts using different network architectures for 10 different runs. Networks have different number of layers, from 1 to 10. Contexts are present in all layers. The white background denotes the absence of a layer.

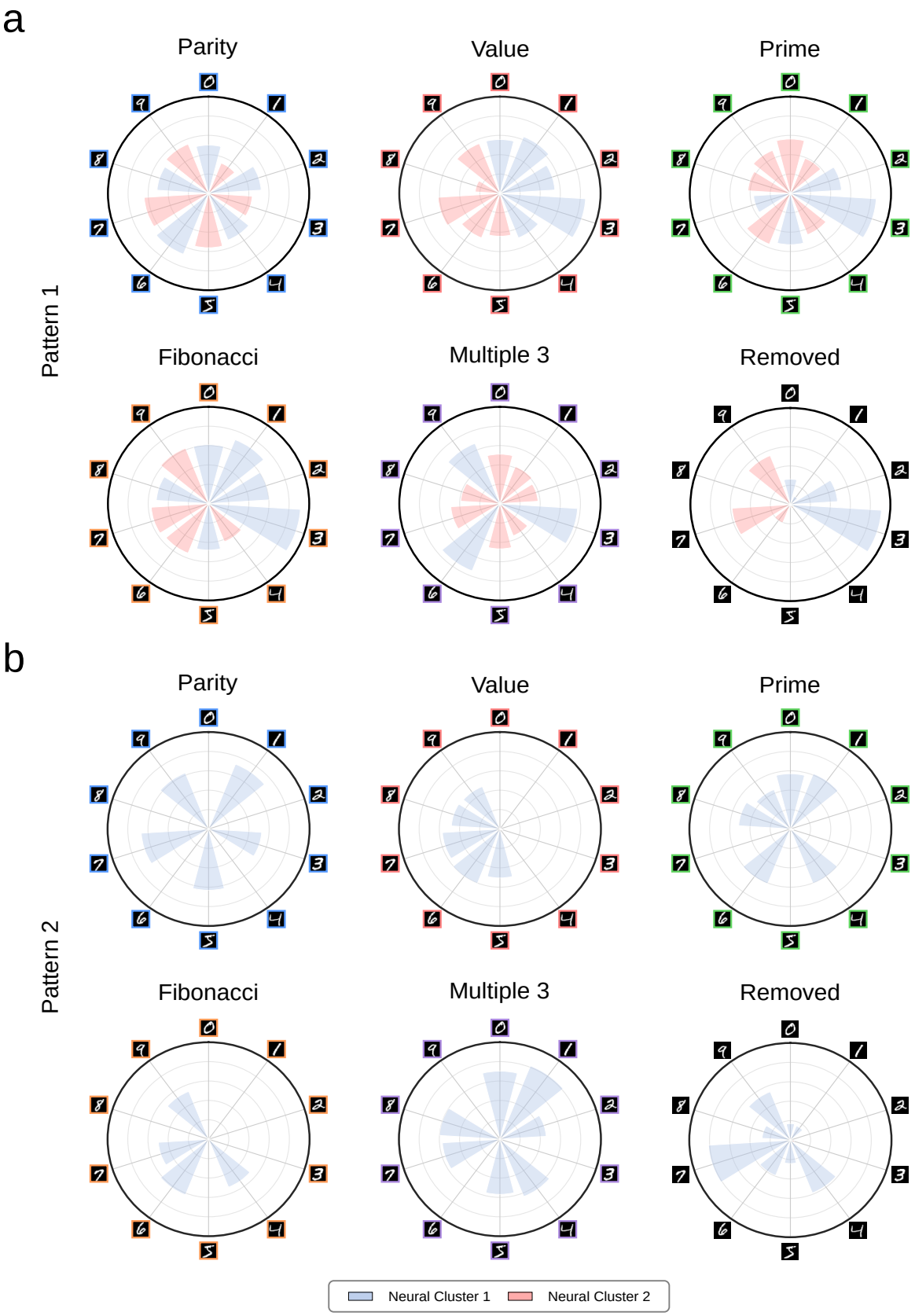

Figure S12: Neural clustering for the {*parity, value, fibonacci, prime, multiple of 3*} *task-switching* schedule. (a) Pattern 1. (b) Pattern 2.