Capsule Networks as Recurrent Models of Grouping and Segmentation

Adrien Doerig[a,†,*], Lynn Schmittwilken[a,†], Bilge Sayim[b,c], Mauro Manassi[d] & Michael H. Herzog[a]

[a] Laboratory of Psychophysics, Brain Mind Institute, EPFL, Lausanne, 1015, Switzerland

[b] Institute of Psychology, University of Bern, 3012 Bern, Switzerland

[c] Univ. Lille, CNRS, UMR 9193- SCALab- Sciences Cognitives et Sciences Affectives, F-59000 Lille, France

[d] Department of Psychology, University of Aberdeen, Aberdeen, Scotland, UK

† Equal contributions

* Corresponding author: adrien.doerig@gmail.com

Keywords: Vision, Neural Networks, Capsule Networks, Crowding, Global Shape Processing, Recurrent Processing

Abstract

Classically, visual processing is described as a cascade of local feedforward computations. Feedforward Convolutional Neural Networks (ffCNNs) have shown how powerful such models can be. However, using visual crowding as a well-controlled challenge, we previously showed that no classic model of vision, including ffCNNs, can explain human global shape processing (1). Here, we show that Capsule Neural Networks (CapsNets; 2), combining ffCNNs with recurrent grouping and segmentation, solve this challenge. We also show that ffCNNs and standard recurrent CNNs do not, suggesting that the grouping and segmentation capabilities of CapsNets are crucial. Furthermore, we provide psychophysical evidence that grouping and segmentation are implemented recurrently in humans, and show that CapsNets reproduce these results well. We discuss why recurrence seems needed to implement grouping and segmentation efficiently. Together, we provide mutually reinforcing psychophysical and computational evidence that a recurrent grouping and segmentation process is essential to understand the visual system and create better models that harness global shape computations.

Author Summary

Feedforward Convolutional Neural Networks (ffCNNs) have revolutionized computer vision and are deeply transforming neuroscience. However, ffCNNs only roughly mimic human vision. There is a rapidly expanding body of literature investigating differences between humans and ffCNNs. Several findings suggest that, unlike humans, ffCNNs rely mostly on local visual features. Furthermore, ffCNNs lack recurrent connections, which abound in the brain. Here, we use visual crowding, a well-known psychophysical phenomenon, to investigate recurrent computations in global shape processing. Previously, we showed that no model based on the classic feedforward framework of vision can explain global effects in crowding. Here, we show that Capsule Neural Networks (CapsNets), combining ffCNNs with recurrent grouping and segmentation, solve this challenge. ffCNNs and recurrent CNNs with lateral and top-down recurrent connections do not, suggesting that grouping and segmentation are crucial for human-like global computations. Based on these results, we hypothesize that one computational function of recurrence is to efficiently implement grouping and segmentation. We provide psychophysical evidence that, indeed, grouping and segmentation is based on time consuming recurrent processes in the human brain. CapsNets reproduce these results too. Together, we provide mutually reinforcing computational and psychophysical evidence that a recurrent grouping and segmentation process is essential to understand the visual system and create better models that harness global shape computations.

Introduction

The visual system is often seen as a hierarchy of local feedforward computations (3), going back to the seminal work of Hubel and Wiesel (4). Low-level neurons detect basic features, such as edges. Pooling these outputs, higher-level neurons detect higher-level features such as corners, shapes, and ultimately objects. Feedforward Convolutional Neural Networks (ffCNNs) embody this classic framework of vision and have shown how powerful it can be (e.g., 5–8). However, ffCNNs only roughly mimic human vision as a large body of literature shows. For example, ffCNNs lack the abundant recurrent processing of humans (9, 10), perform differently than humans in crucial psychophysical tasks (1, 11), and can be easily misled (12–14). An important point of discussion concerns global visual processing. It was suggested that ffCNNs mainly focus on local, texture-like features, while humans harness global shape computations (1, 14–18; but see 19)). In this context, it was shown that changing local features of an object, such as its texture or edges, leads ffCNNs to misclassify (14, 15), while humans can still easily classify the object based on its global shape.

There are no widely accepted diagnostic tools to specifically characterize global computations in neural networks. Models are usually compared on computer vision benchmarks, such as ImageNet (20), or with neural responses in the visual system (21, 22). One drawback of these approaches is that the datasets are hard to control. Psychophysical results can be used to fill this gap and create well-controlled challenges for visual models, tailored to target specific aspects of vision (23). Here, we use visual crowding to specifically target global shape computations in humans and artificial neural networks.

In crowding, objects that are easy to identify in isolation appear as jumbled and indistinct when clutter is added (1, 24–29). Consider the following example: Figure 1a shows a vernier target, i.e., two vertical lines separated by a horizontal offset. When the vernier is presented alone, observers easily discriminate the offset direction. When a flanking square surrounds the target, performance drops, i.e., there is strong crowding (30, 31). Surprisingly, *adding* more flankers can *reduce* crowding strongly, depending on the spatial configuration of the flankers (Figure 1b; 29). Hence, the *global* configuration of visual elements across large regions of the visual field influences perception of the small vernier target. This global *un*crowding effect occurs for a wide range of stimuli in vision, including foveal and peripheral vision, audition, and haptics (32–38). The ubiquity of (un)crowding in perception is not surprising, since visual elements are rarely seen in isolation. Hence, any perceptual system needs to cope with crowding to isolate important information from clutter.

Previously, we have shown that these global effects of crowding *cannot* be explained by models based on the classic framework of vision, including ffCNNs (1, 18, 39). Here, we propose a new framework to understand these global effects. We show that Capsule Neural Networks (CapsNets; 2), augmenting ffCNNs with a recurrent grouping and segmentation process, can explain these complex global (un)crowding results in a natural manner. Two processing regimes can occur in CapsNets: a fast feedforward pass that is able to quickly process information, and a time-consuming recurrent regime to compute in-depth global grouping and segmentation. We will show psychophysical evidence that the human visual system indeed harnesses recurrent processing for efficient grouping and segmentation, and that CapsNets naturally reproduce this behavior as well. Together, our results suggest that a time-consuming recurrent grouping and segmentation process is crucial for global shape computations in both humans and artificial neural networks.

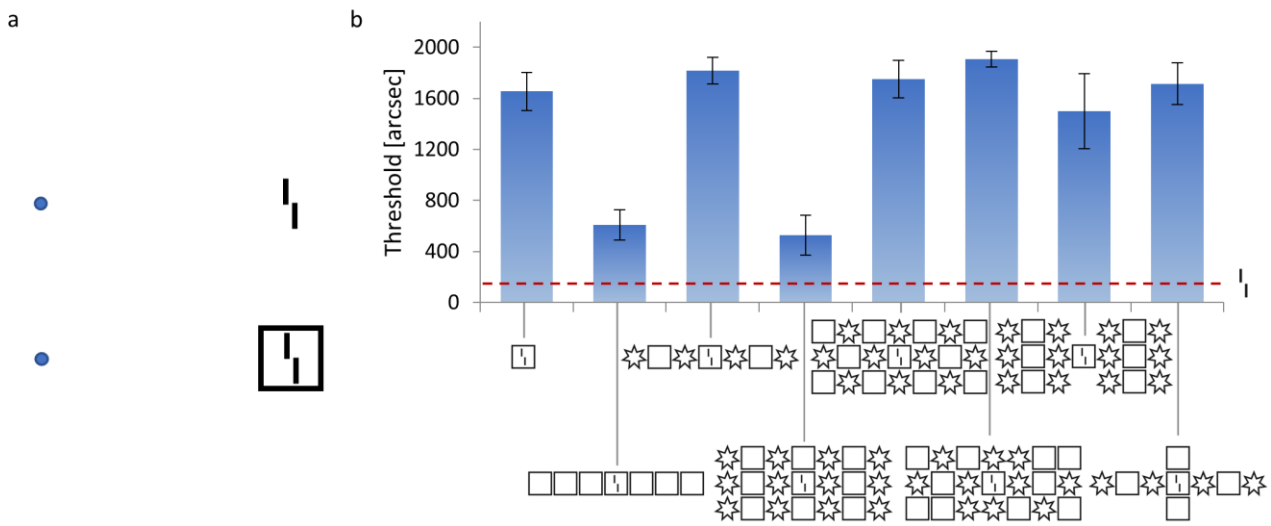


Figure 1: (a) Crowding: The perception of visual elements deteriorates in clutter, an effect called crowding. When fixating on the blue dots, the vernier (i.e., two vertical bars with a horizontal offset) becomes harder to perceive when it is flanked by a square. **(b) Uncrowding:** Results of a psychophysical experiment in which stimuli are presented in the visual periphery. The y-axis shows the minimal offset size at which observers can report the offset direction with 75% accuracy (i.e., lower values indicate better performance). The offset direction of a vernier in isolation can be easily reported (dashed red line). When a flanking square surrounds the vernier, performance deteriorates (crowding). When more squares are added, performance recovers (uncrowding). Critically, the uncrowding effect depends on the global stimulus configuration. For example, if some squares are replaced by stars, performance deteriorates again (3rd bar; 29).

Results

Experiment 1: Crowding and Uncrowding Naturally Occur in CapsNets

In CapsNets, early convolutional layers extract basic visual features. Recurrent processing combines these features into groups and segments objects by a process called *routing by agreement*[1]. The entire network is trained end-to-end through backpropagation. *Capsules* are groups of neurons representing visual features and are crucial for the routing by agreement process. Low-level capsules iteratively predict the activity of high-level capsules in a recurrent loop. If the predictions agree, the corresponding high-level capsule is activated. For example, if a low-level capsule detects a rectangle, and the capsule above detects a triangle, as shown in Figure 2a, they agree that the higher-level object should be a house and, therefore, the corresponding high-level capsule is activated (Figure 2a). This process allows CapsNets to group and segment objects (Figure 2b). Because of these capabilities, we

[1] In most implementations of CapsNets, including ours and (2), the iterative routing by agreement process is not explicitly implemented as a "standard" recurrent neural network processing sequences of inputs online. Instead, there is an iterative algorithmic loop (see (2) for the algorithm), which is equivalent to recurrent processing.

hypothesized that CapsNets are able reproduce human-like (un)crowding in a visual crowding experiment.

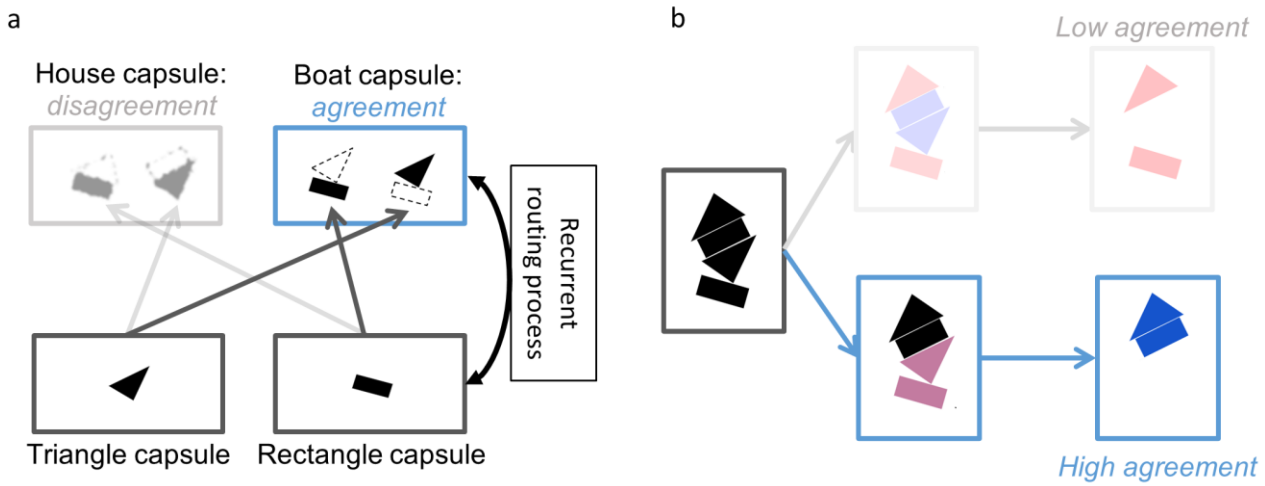


Figure 2: (a) Routing by agreement in CapsNets: A capsule is a group of neurons whose activity vector represents an entity as well as its instantiation parameters (such as position, orientation, color etc.). In this example, lower-level capsules detect triangles and rectangles as well as their orientations. Higher-level capsules detect combinations of these shapes. Here, the triangle capsule detects a tilted triangle and the rectangle capsule detects a tilted rectangle. Then, each capsule predicts the representation of the higher-level capsule. Based on the orientation of its own representation, the triangle capsule predicts an upside-down house and a tilted boat. Similarly, the rectangle capsule predicts a tilted house and a tilted boat. In CapsNets, information between layers gets propagated recurrently in order to maximize their agreement. In this case, capsules agree on the orientation of the boat but disagree on the orientation of the house. Hence, the routing by agreement process suppresses activity in the house capsule and boosts activity in the boat capsule. (**b) Grouping and segmentation in CapsNets:** The recurrent routing by agreement process endows CapsNets with natural grouping and segmentation capabilities. Here, an ambiguous stimulus is presented, which can either be seen as an upside-down house (top) or a house and a boat (bottom). The upside-down house interpretation leaves parts of the image unexplained and this causes disagreement. Routing by agreement will boost the bottom interpretation, because it leads to most agreement in the network. In order to do so, the individual features get grouped into a house and a boat and segmented into the corresponding higher-level capsules.

We trained CapsNets with two convolutional layers followed by two capsule layers to recognize greyscale images of vernier targets and groups of identical shapes (see Methods). During training, either a vernier or a group of identical shapes was presented. The network had to simultaneously classify the shape type, the number of shapes in the group, and the vernier offset direction. Importantly, verniers and shapes were never presented together during training. Therefore, the network was never exposed to (un)crowding stimuli during training.

During testing, we evaluated the vernier offset discrimination of the networks when presented with (un)crowding stimuli. The CapsNets reproduce both crowding and uncrowding as in psychophysical experiments (Figure 3a): presenting the vernier within a single flanker deteriorated offset discrimination performance (crowding). Adding more identical flankers recovered performance (uncrowding). Adding configurations of alternating different flankers did not recover performance (crowding), similarly to human vision. Small changes in the network hyperparameters, loss terms or stimulus characteristics did not affect these results (supplementary material). As a control condition, we checked that when the vernier target is presented outside the flanker configuration, rather than inside, there was no performance drop (supplementary material). Hence, the performance drop in crowded conditions was not merely due to the simultaneous presence of the target and flanking shape in the stimulus, but really comes from crowding between the vernier and flankers.

Because each feature is represented in its own capsule, we can decode how each feature of a stimulus is represented in its capsule to gain important insights about the underlying computations. This is an advantage compared to ffCNNs, in which it is unclear which neurons represent which feature. Reconstructing the input images based on the network's output (see Methods) shows that the difference between crowding and uncrowding comes from grouping and segmentation (Figure 3b). More specifically, crowding occurs when the target and flankers cannot be segmented and are therefore routed to the same capsule. In this case, they interfere because a single capsule cannot represent well two objects simultaneously due to limited neural resources. This mechanism is similar to pooling: information about the target is pooled with information about the flankers, leading to poorer representations. However, if the flankers are segmented away and represented in a different capsule, the target is released from the flankers' deleterious effects and *un*crowding occurs (Figure 3c). This segmentation can only occur if the network has learnt to group the flankers into a single higher-level object represented in a different capsule than the vernier. Segmentation is facilitated when more flankers are added because more capsules agree on the presence of the flanker group.

Alternating configurations of different flankers, as in the third configuration of Figure 1b, usually do not lead to uncrowding in humans (29). In some rare cases, our CapsNets produced uncrowding with such configurations (e.g. configurations u, C and D in Figure 3a). Reconstructions suggest that in these cases the network could not differentiate between the different shapes of the flankers (e.g. between diamonds and stars). Therefore, the flankers formed a group for the network and were segmented away from the target (Figure 3b). This further reinforces the notion that grouping and segmentation differentiate crowding from uncrowding: whenever the network reaches the conclusion that the

flankers form a group, segmentation is facilitated. When this happens, the vernier and flankers are represented in different capsules and therefore do not interfere, leading to good performance.

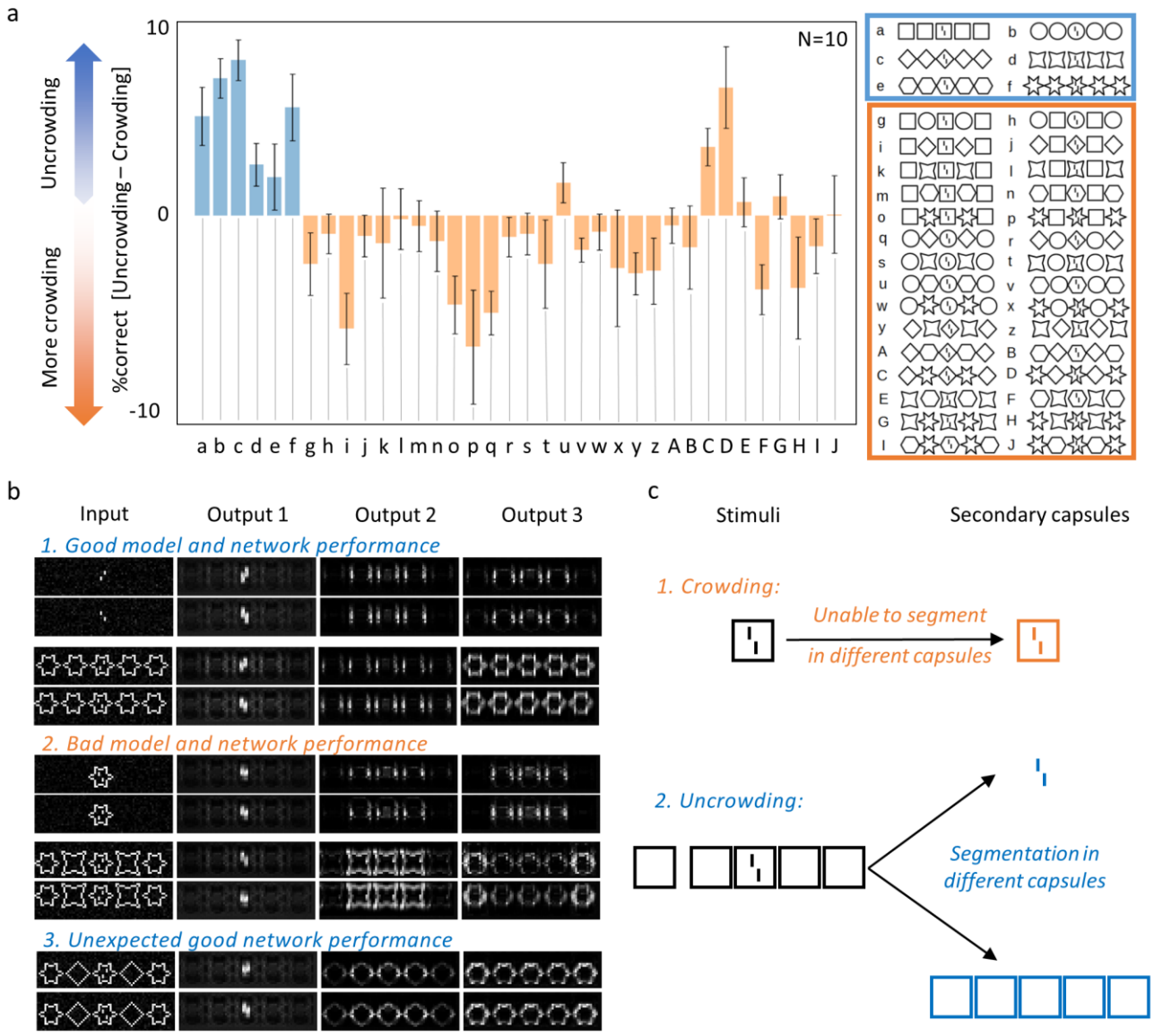


Figure 3: a. CapsNets explain both crowding and uncrowding: The x-axis shows the configurations that were used to test (un)crowding. We used 6 different flanker shape types and tested all configurations with 5 identical or alternating shapes (e.g., 5 squares, 5 circles, circle-square-circle-square-circle, etc; see Methods). Performance is shown on the y-axis as %correct for the whole configuration *minus* %correct for the central flanker alone. For example, in column *a*, vernier offset discrimination is better with 5 square flankers than with 1 square flanker. We trained N=10 networks to match the typical number of observers in human experiments (29, 40). Error bars indicate the standard error across all networks. The blue bars represent configurations for which *un*crowding is expected. Values larger than 0 are in accordance with human data. Orange bars represent configurations for which crowding is expected. Values smaller than or equal to 0 are in accordance with human data. **b. Reconstructions:** We reconstructed the input images based on the secondary capsules' activities (see Methods). The reconstructions from the three most active secondary capsules are shown. When the vernier

is presented alone (1. top), the reconstructions are good. When a single flanker is added (panel 2 top), the vernier and flanker interfere because the vernier is not well segmented from the flanker, and reconstructions deteriorate (crowding). Sometimes, the vernier is reconstructed with an incorrect offset. When identical flankers are added (panel 1 bottom), both the vernier and the flanker reconstructions recover, because the vernier is well segmented from the flankers (uncrowding). With different flankers (2. bottom), the segmentation is unsuccessful and reconstructions deteriorate again (crowding). Sometimes, the vernier is reconstructed with the incorrect offset. Interestingly, the reconstructions in case of "unexpected" uncrowding (i.e., the networks show uncrowding contrary to humans) often resemble the ones in case of "normal" uncrowding: one flanker type is well segmented and reconstructed (compare panel 3 with panels 1&2 bottom). This suggest that the network was unable to differentiate between diamonds and stars, and treated this configuration as a large group of stars. **c. Segmentation and (un)crowding in CapsNets:** If Caps-Nets can segment the vernier target away from the flankers during the recurrent routing by agreement process, uncrowding occurs. Segmentation is difficult when a single flanker surrounds the target because capsules disagree about what is shown at this location. In the case of configurations that the network has learnt to group, many primary capsules agree about the presence of a group of shapes. Therefore, they can easily be segmented away from the vernier.

In previous work, we have shown that pretrained ffCNNs cannot explain uncrowding (18), even if it is biased towards global shape processing (14). Currently, CapsNets cannot be trained on large-scale tasks such as ImageNet because routing by agreement is computationally too expensive. Therefore, we took a different approach here. As explained above, we trained our CapsNets to recognize groups of shapes and verniers and asked how they would generalize from shapes presented in isolation to crowded shapes. To make sure that CapsNets explain global (un)crowding due to their *architecture* focusing on grouping and segmentation and not merely due to a difference in the *training regime*, we conducted three further experiments. We investigated how vernier discrimination performance changes when the capsule layers are replaced by other architectures, keeping the number of neurons constant.

First, we replaced the capsule layers by a fully connected feedforward layer, yielding a standard ffCNN with three convolutional layers and a fully connected layer. We trained and tested this architecture in the same way as the CapsNets. The results clearly show that there is no uncrowding (Figure 4a): ffCNNs do not reproduce human-like global computations with this procedure.

Second, we added lateral recurrent connections to the fully connected layer of the ffCNN, yielding a network with three convolutional layers followed by a fully connected recurrent layer. We used the same number of recurrent iterations as for the routing by agreement in the CapsNets, and trained the network the same way as before. There is no uncrowding with this architecture either (Figure 3b).

Lastly, we added top-down recurrent connections feeding back from the final fully connected layer of the ffCNN to the layer below, yielding a network with three convolutional layers followed by a fully connected layer that fed back into the previous layer (again with the same number of recurrent iterations as iterations of routing by agreement in the CapsNets). After the same training, this architecture did not produce any uncrowding either (Figure 4c).

The absence of uncrowding in ffCNNs and recurrent CNNs with lateral or top-down recurrent connections suggests that the *architecture* of CapsNets, and not our *training regime* explains why (un)crowding is reproduced. Furthermore, recurrent processing by itself is not sufficient to produce (un)crowding. The grouping and segmentation performed by routing by agreement seems crucial.

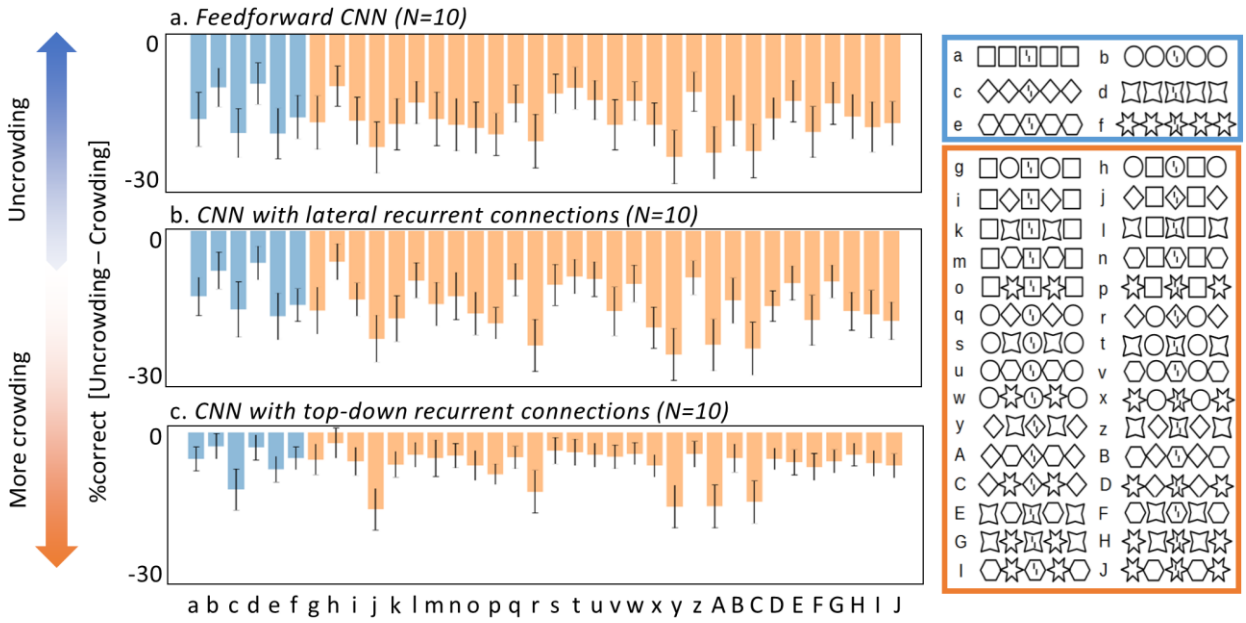


Figure 4: Other network architectures do not explain uncrowding. To verify that the ability of CapsNets to explain uncrowding is due to their architecture and not merely to the way they are trained, we replaced the recurrent routing by agreement between capsules by three different alternative architectures: a feedforward fully connected layer (yielding a classic ffCNN, a), a fully connected layer with lateral recurrent connections (b) and a fully connected layer with top-down recurrent connections to the layer below (c). The x-axes represent the (un)crowding configurations used during testing (shown on the right). The y-axes show the vernier discrimination performance as %correct for the whole configuration minus %correct for the central flanker alone as in Figure 3a. None of the ffCNN or recurrent CNN architectures we tested can produce uncrowding (compare with the CapsNet results in figure 3a).

Experiment 2: The role of recurrent processing

Processing in CapsNets starts with a feedforward sweep which is followed by a time-consuming recurrent routing by agreement process to refine grouping and segmentation. We hypothesized that

the human brain may use a similar strategy to efficiently implement grouping and segmentation. To test this hypothesis, we psychophysically investigated the temporal dynamics of (un)crowding in humans.

For this, we performed a psychophysical crowding experiment with a vernier target flanked by either two lines or two cuboids (see Methods; Figure 5). The stimuli were displayed for varying durations from 20 to 640ms. The observers had to report the vernier offset direction. For short stimulus durations, crowding occurred for both flanker types, i.e., vernier offset thresholds were significantly larger in both the lines condition and cuboids conditions compared to the vernier alone condition (lines: $p = 0.0017$, cuboids: $p = 0.0013$, 2-tailed one-sample t-tests).

To quantify how performance changed with increasing stimulus duration, we fitted a line $y = ax + b$ to the data for each subject, and compared the slopes a between the lines condition and the cuboids condition. Discrimination performance in the cuboids condition improved significantly more with increasing stimulus duration than performance in the lines condition ($p=0.003$, 2-tailed 2-sample t-test). Previous results have suggested that crowding varies little with stimulus duration (41; but see 42, 43). As shown in Figure 5, this may be the case for the lines condition of our experiment, but *not* for the cuboids condition, where performance drastically improves with stimulus duration. This difference in performance between the lines and cuboids conditions cannot be explained by local mechanisms such as lateral inhibition (30, 44) or pooling (45–47), since the inner vertical lines are the same for the line and cuboid flankers.

Crucially, uncrowding occurred for the cuboid flankers only when stimulus durations were sufficiently long (Figure 4). We propose that these results reflect the time-consuming recurrent computations needed to segment the cuboid flankers away from the target. Performance does not improve as much with the line flankers, because they are too strongly grouped with the vernier target, so recurrent processing cannot segment them away.

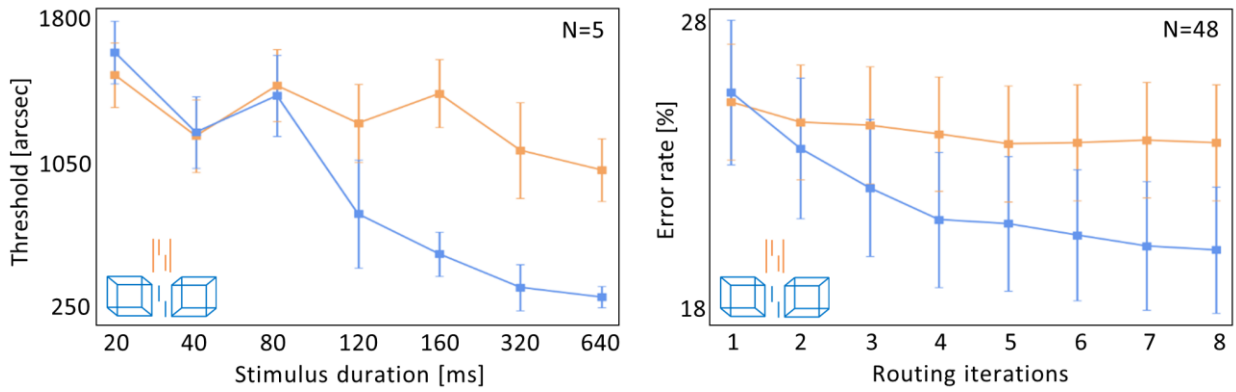


Figure 5: Temporal dynamics of uncrowding: *Left: Human data*. The x-axis shows the stimulus durations and the y-axis shows the corresponding thresholds (i.e., lower values indicate better performance). Error bars indicate the standard error. For cuboid flankers, crowding occurs up to 100ms of stimulus presentation, but performance gradually improves with increasing stimulus duration (i.e., uncrowding occurs with time, blue). In contrast, there is no uncrowding with increasing stimulus duration for the line flankers (orange). This suggests that the cuboids are segmented from the vernier through time-consuming recurrent processing. In contrast, the line flankers are strongly grouped with the vernier and cannot be segmented at all. *Right: Model data.* CapsNets can explain these results by varying the number of recurrent routing by agreement iterations. The x-axis shows the number of recurrent routing iterations and the y-axis shows the error rates for the vernier discrimination performance (i.e., lower values indicate better performance). Error bars indicate the standard error across all N=48 trained networks (see Methods). CapsNets reproduce the results of the human psychophysical experiment: discrimination errors decrease with an increasing number of recurrent routing iterations, but only in the cuboids condition. This suggests that recurrent processing helps to compute and segment the cuboids, but the lines are too strongly grouped with the vernier to be segmented. Hence, they do not benefit from the recurrent segmentation process. Note that the blue and orange stimulus colors are for illustration only. All stimuli were presented on the same oscilloscope and had the same color.

To investigate if CapsNets reproduce these results, we trained CapsNets with the same architecture as in experiment 1 to discriminate vernier offsets, and to recognize lines, cuboids and scrambled cuboids (see Methods). The scrambled cuboids were included to prevent the network from classifying lines vs. cuboids simply based on the number of pixels in the image. As in experiment 1, each training sample contained only one of the shape types. We used 8 routing by agreement iterations during training. As in experiment 1, verniers and flankers were never presented together during training (i.e., there were no (un)crowding stimuli).

After training, we tested the networks on (un)crowding stimuli, changing the number recurrent routing by agreement iterations from one (leading to a purely feedforward regime) to 8 iterations (a highly recurrent regime; Figure 5). We found that CapsNets naturally explain the human results. Using the

same statistical analysis as for the psychophysical experiment, we found that discrimination performance improves significantly more with increasing routing iterations in the cuboids condition as compared to the lines condition ($p=0.007$; 2-tailed 2-sample t-test). As shown in Figure 5, additional processing time only has a small effect on performance for the line stimuli. In contrast, with more iterations the cuboids are better segmented from the target, and performance improves strongly. These results were not affected by small changes in network hyperparameters or loss terms (supplementary material). We did not compare these results with the ffCNN and recurrent networks used in experiment 1, because these networks produced no uncrowding at all.

These findings are explained by the recurrent routing by agreement process. With cuboids, capsules across an extended spatial region need to agree about the presence of a cuboid, which is then segmented into its own capsule. This complex process requires several recurrent iterations of the routing by agreement process. On the other hand, the lines are much more strongly grouped with the vernier, so further iterations of routing by agreement achieve worse segmentation and, hence, cannot improve performance as much.

Discussion

Our results provide strong evidence that time-consuming recurrent grouping and segmentation is crucial for shape-level computations in both humans and artificial neural networks. We used (un)crowding as a psychophysical probe to investigate how the brain flexibly forms object representations. These results specifically target global, shape-level and time-consuming recurrent computations and constitute a well-controlled and difficult challenge for neural networks.

It is well known that humans can solve a number of visual tasks very quickly, presumably in a single feedforward pass of neural activity (48). ffCNNs are good models of this kind of visual processing (21, 22, 49). However, many studies have shown that neural activities are not determined by the feedforward sweep alone, and recurrent activity affords a distinct processing regime to perform more in-depth and time-consuming computations (9, 10, 50–53). Similarly, CapsNets naturally include both a fast feedforward and a time-consuming recurrent regime. When a single routing by agreement iteration is used, CapsNets are rapid feedforward networks that can accomplish many tasks, such as vernier discrimination or recognizing simple shape types (e.g. circles vs. squares). With more routing iterations, a recurrent processing regime arises and complex global shape effects emerge, such as segmenting the cuboids in experiment 2. We showed how the transition from feedforward to

recurrent processing in CapsNets explains psychophysical results about temporal dynamics of (un)crowding.

Recurrent activity offers several advantages. First, although feedforward networks can in principle implement any function (54), recurrent networks can implement certain functions more efficiently. Flexible grouping and segmentation is exactly the kind of function that may benefit from recurrent computations (see also 55). For example, to determine which local elements should be grouped into a global object, it helps to compute the global object first. This information can then be fed back to influence how each local element is processed. For example, to model (un)crowding, it helps to compute the global configuration of flankers first to determine how to process the vernier. Should it be grouped with the flankers (crowding) or not (uncrowding)? In CapsNets, the first feedforward sweep of activity provides an initial guess about which global objects are present (e.g., large cuboids). At this stage, as shown in experiment 2, information about the vernier interferes with information about the cuboids (crowding). Then, recurrent processing helps to route the relevant information about the cuboids and the vernier to the corresponding capsules (uncrowding). Without recurrence, it is difficult to rescue the vernier information once it has been crowded.

Second, although any network architecture can implement any computation in principle (given enough neurons), they differ in the way they *generalize* to previously unseen stimuli. Hence, recurrent grouping and segmentation architectures influence what is learnt from the training data. Here, we have shown that only CapsNets, but not ffCNN or CNNs augmented with recurrent lateral or top-down connections, produce uncrowding when trained identically to recognize groups of shapes and verniers. In general, ffCNNs tend to generalize poorly (review: 56)). Using different architectures to improve how current systems generalize is a promising avenue of research. In this respect, we have shown that CapsNets generalize more similarly to humans than ffCNNs and standard recurrent networks in the context of global (un)crowding. Relatedly, CapsNets may also be more resilient to adversarial attacks (57, 58), generalize better to new viewpoints (59), and outperform ffCNNs on difficult visual tasks with scarce training data such as tumor detection (60), also suggesting potentially more human-like generalization than ffCNNs.

One limitation in our experiments is that we explicitly taught the CapsNets which configurations to group together by selecting which groups of shapes were present during training (e.g., only groups of identical shapes in experiment 1). Effectively, this gave the network adequate priors to produce un-crowding with the appropriate configurations (i.e., only identical, but not different flankers). Hence, our results show that, given adequate priors, CapsNets explain uncrowding. We have shown that

ffCNNs and CNNs with lateral or top-down recurrent connections do *not* produce uncrowding, *even* when they are trained identically on groups of identical shapes and showed learning on the training data comparable to the CapsNets (furthermore, we showed previously that ffCNNs trained on large datasets who are often used as general models of vision do not show uncrowding either; 18). This shows that merely training networks on groups of identical shapes is not sufficient to explain uncrowding. It is the recurrent segmentation in CapsNets that is crucial. Humans do not start from zero and therefore do not need to be trained in order to perform crowding tasks. The human brain is shaped through evolution and learning to group elements in a useful way to solve the tasks it faces. As mentioned, (un)crowding can be seen as a probe into this grouping strategy. Hence, we expect that training CapsNets on more naturalistic tasks such as ImageNet may lead to grouping strategies similar to humans and may therefore naturally equip the networks with priors that explain (un)crowding results. At the moment, however, CapsNets have not been trained on such difficult tasks because the routing by agreement algorithm is computationally too expensive.

The approach we took in this contribution goes beyond the standard use of deep learning in neuroscience, in which large ffCNN are fitted on large datasets to model behavior. To our knowledge, there is little research on the *specific* kinds of computations that ffCNNs *cannot* do and, importantly, *why they cannot do it*. Here, we have shown that a very specific kind of recurrent computations is needed to explain global processing: recurrent grouping and segmentation. Furthermore, experiment 2 goes beyond the usual static image recognition measures by taking time-extended computations into account. There are currently very few attempts to explain the dynamics of human neural computations using deep learning (with the notable exception of (10)). This may be because most researchers use ffCNNs, which are ill-suited to address such dynamic computations. Contrasting different non-standard deep learning architectures to ask specific neuroscientific questions about well-defined tasks can lead to more precise ways of using deep networks in neuroscience. We think such approaches will be important in the future.

Recurrent networks are harder to train than feedforward systems, which explains the dominance of the latter during these early days of deep learning. However, despite this hurdle, recurrent networks are emerging to address the limitations of ffCNNs as models of the visual system (10, 50, 52, 53, 61, 62). Although there is consensus that recurrence is important for brain computations, it is currently unclear which functions exactly are implemented recurrently, and how they are implemented. Our results suggest that one important role of recurrence is global shape-level computation through grouping and segmentation. We had previously suggested another recurrent segmentation network,

hard-wired to explain uncrowding (63). However, CapsNets, bringing together recurrent grouping and segmentation with the power of deep learning, are much more flexible and can be trained to solve any task. Linsley et al. (53) proposed another recurrent deep neural network for grouping and segmentation, and there are other possibilities too (64, 65). We do not suggest that CapsNets are the only implementation of grouping and segmentation. We only suggest that grouping and segmentation is important and CapsNets are one potential way in which grouping could be solved. Further work is needed to show how the brain implements it. Regardless of the specific implementation, a focus on recurrent grouping and segmentation offers a fresh way of conceptualizing global visual processing and has important implications far beyond crowding, for neuroscience and computational science in general. Harnessing the power and flexibility offered by deep learning approaches beyond classic ffCNNs will be crucial to understand and model these complex processes. We suggest that important new avenues of research will open up when complex deep networks such as CapsNets can be scaled to deal with large complex datasets, and this contribution is an example of this promising approach.

In conclusion, our results provide mutually reinforcing modelling and psychophysical evidence that time-consuming, recurrent grouping and segmentation plays a crucial role for global shape computations in humans and machines.

Methods

The code to reproduce our results is available at https://github.com/adriendoerig/Capsule-networks-as-recurrent-models-of-grouping-and-segmentation. All models were implemented in Python 3.6, using the high-level estimator API of Tensorflow 1.10.0. Computations were run on a GPU (NVIDIA GeForce GTX 1070).

We used the same basic network architecture in all experiments, motivated by the following rationale (Figure 6a). After training, ideally, primary capsules detect the individual shapes present in the input image, and the secondary capsules group and segment these shapes through recurrent routing by agreement. The network can only group shapes together because it was taught during training that specific shapes form a group. To match this rationale, we set the primary capsules' receptive field sizes to roughly match the size of one shape.

Early feature extraction was implemented with two convolutional layers without padding, each followed by an ELU non-linearity. These convolutional layers were followed by two capsule layers using routing by agreement. The primary capsules were implemented by a third convolution layer, reshaped

into *m* primary capsule types outputting *n*-dimensional activation vectors (this is the standard way to implement primary capsules). Finally, the number of secondary capsule types was equal to the number of different shapes used as input. Different decoders were implemented to measure different loss terms (Eqs. 1-7). Each decoder used the secondary capsules as inputs. The networks were trained end-to-end through backpropagation. For training, we used an Adam optimizer with a batch size of 48 and a learning rate of 0.0004. To this learning rate, we applied cosine decays with warm restarts (62). Supplementary table 1 summarizes all parameters of the network. Results for different parameters are shown in the supplementary material. In general, small changes in the network parameters did not affect results.

Inputs were grayscale images with different shapes (Figure 7 and 8). We added random Gaussian noise with mean $\mu = 0$ and a standard deviation randomly drawn from a uniform distribution. The contrast was varied either by first adding a random value between -0.1 and 0.1 to all pixel values and then multiplying them with a contrast adjustment factor randomly drawn from a uniform distribution or vice versa. The pixel values were clipped between 0 and 1.

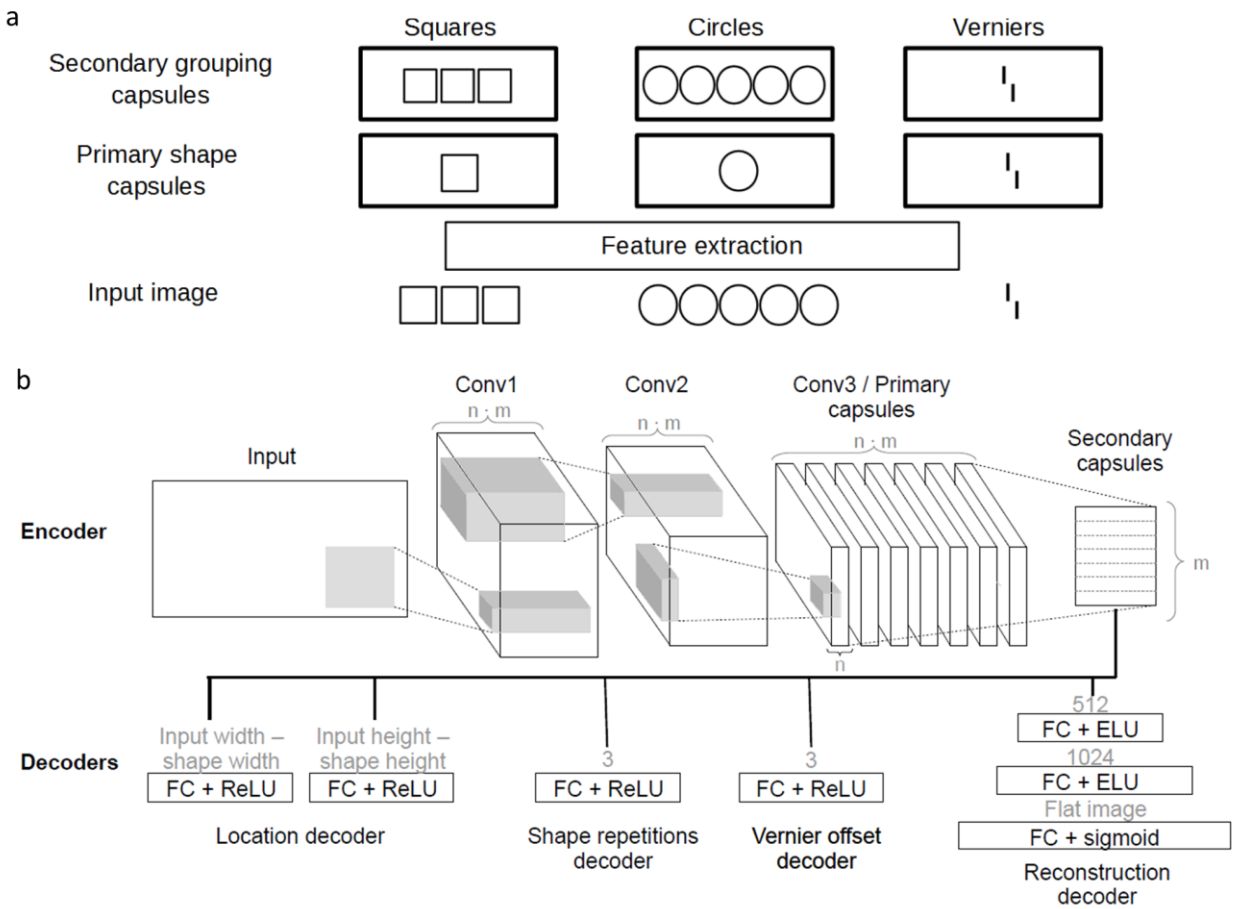


Figure 6: (a) Ideal representations: After training, we expected the primary capsules to detect single shapes of different types (here: squares, circles and verniers), and secondary capsules to group these shapes in group of 1, 3, or 5. If 3 squares

are presented, the primary square capsules detect squares at 3 locations. Through routing by agreement, the secondary squares capsule detects this group of 3 squares. If 5 circles are presented, the primary circle capsules detect circles at 5 locations. After routing, the secondary circles capsule represents a group of 5 circles. If a vernier is presented, it is detected and routed to the secondary vernier capsule. **(b) Network architecture:** We used capsule networks with two convolutional layers, one primary capsule layer with m primary capsule types and n primary capsule dimensions, and one secondary capsule layer with m secondary capsule types. The primary capsules were implemented by a third convolution layer, reshaped into m primary capsule types outputting n-dimensional activation vectors (this is the standard way to implement primary capsules). In this example, there are seven primary and secondary capsules types to match the seven shape types used in experiment 1 (see caption a). The primary and secondary capsule layers communicate via routing-by-agreement. Small fully connected decoders are applied to the secondary capsules to compute different loss terms (see main text). The network is trained end-to-end through backpropagation to minimize all losses simultaneously. We used different network hyperparameters in the main text results and in the supplementary material to ensure that our results are stable against hyperparameter changes (see supplementary Table 1 for all hyperparameter values).

Experiment 1:

Modelling

Human data for experiment 1 is based on (29). We trained CapsNets with the above architecture to solve a vernier offset discrimination task and classify groups of identical shapes. The loss function included a term for shape type classification, a term for vernier offset discrimination, a term for the number of shapes in the image, a term for reconstructing the input based on the network output, and a term localizing the stimulus position in the image (see equations 1-6). The shape repetition, reconstruction and location loss terms are used to stabilize performance but are not essential for producing (un)crowding (see supplementary material). Each loss term was scaled so that none of the terms dominated the others. For the shape type classification loss, we implemented the same margin loss as in (2). This loss enables the detection of multiple objects in the same image. For the vernier offset loss, we used a small decoder to determine vernier offset directions based on the activity of the secondary vernier capsule. The decoder was composed of a single dense hidden layer followed by a ReLU-nonlinearity and a dense readout layer of three nodes corresponding to the labels left and right, or no vernier. The vernier offset loss was computed as the softmax cross entropy between the decoder output and the one-hot-encoded vernier offset labels. The loss term for the number of shapes in the image was implemented similarly, but the output layer comprised three nodes representing the labels one, three or five shape repetitions. For the reconstruction loss, we trained a decoder with two fully connected hidden layers (h1: 512 units, h2: 1024 units) each followed by ELU nonlinearities to reconstruct the input image. The reconstruction loss was calculated as the squared

difference between the pixel values of the input image and the reconstructed image. The total loss is given by the following formulas:

$$L_{total} = \alpha_{shape\ type} L_{shape\ type} + \alpha_{vernier\ offset} L_{vernier\ offset} + \alpha_{shape\ repetitions} L_{shape\ repetitions} + \alpha_{reconstruction} L_{reconstruction} + \alpha_{location} L_{location} \quad (1)$$

$$L_{shape\ type} = \sum_{k} T_k \max(0, (m^+ - \|v_k\|)^2) + \lambda(1 - T_k) \max(0, (\|v_k\| - m^-)^2) \quad (2)$$

$$L_{vernier\ offset} = \text{xentropy}(\textit{true offset direction}, \textit{predicted offset direction}) \quad (3)$$

$$L_{shape\ repetitions} = \text{xentropy}(\textit{true shape repetition}, \textit{predicted shape repetition}) \quad (4)$$

$$L_{reconstruction} = \sum_{i,j} (\textit{input image}(i,j) - \textit{reconstructed image}(i,j))^2 \quad (5)$$

$$L_{location} = \sum_{i=x,y} \text{Xentropy}(\textit{true coordinate}_i, \textit{predicted coordinate}_i) \quad (6)$$

Where the α are real numbers scaling each loss term, T_k = 1 if shape class k is present, $\|v_k\|$ is the norm of secondary capsule k, and m^+, m^- and λ are parameters of the margin loss with the same values as described in (2). All loss terms decreased during training (see supplementary material).

The training dataset included vernier stimuli and six different shape types (Figure 7a). Shapes were presented in groups of one, three or five shapes of the same type. The group was roughly centered in the middle of the image. We trained N=10 networks and averaged their performance. Importantly, the network was never exposed to (un)crowding stimuli during training. Therefore, the network could not trivially learn when to (un)crowd by overfitting on the training dataset. This situation is similar for humans: they know about shapes and verniers, but their visual system has never been trained on (un)crowding stimuli. After training, we tested the vernier discrimination performance on (un)crowding stimuli (Figure 7b), and obtained reconstructions of the input images.

Figure 7: (a) Training stimuli: All shapes were shown randomly in groups of one, three or five. Verniers were always presented alone. **(b) Testing stimuli:** Example stimuli for the test conditions: In the vernier-alone condition (*left*), we expected

the network to perform well on the vernier discrimination task. In crowding conditions (*middle-left*), we expected a deterioration of the discrimination performance. In uncrowding conditions with identical flankers (*middle-right*), we expected a recovery of the performance. In no-uncrowding conditions with alternating flanker types (*right*), we expected crowding.

To check that CapsNets explain uncrowding because of the grouping and segmentation capabilities offered by routing by agreement and not merely because of the way they are trained, we replaced the capsule layers by other architectures (a feedforward fully connected layer, a fully connected layer with lateral recurrent connections and a fully connected layer with top-down recurrent connections to the layer below; see Results). All these networks had the same number of neurons as our CapsNets, and we used the same number of recurrent iterations as the number of routing by agreement used for the CapsNets. All networks were trained and tested the same way. The only difference is that CapsNets represent different classes in different capsules, so that we could decode information directly from specific capsules. For example, we could decode vernier offsets specifically from the secondary vernier capsule or reconstruct squares specifically from the secondary squares capsule. The standard ffCNNs with and without recurrent connections do not offer this possibility, because different classes are not represented in different known groups of neurons. Therefore, we decoded vernier offsets, reconstructions, the number of shapes and the shape type from the entire last layer of the network rather than from specific capsules. This difference did not limit the networks' performance, since these architectures performed well during training. Hence, the fact that they do not produce uncrowding is not explained by training limitations, but rather by the fact that they *generalize* to novel inputs differently than CapsNets.

Experiment 2:

Psychophysical experiment:

Observers

For experiment 2, we collected human psychophysical data. Participants were paid students of the Ecole Polytechnique Fédérale de Lausanne (EPFL). All had normal or corrected-to-normal vision, with a visual acuity of 1.0 (corresponding to 20/20) or better in at least one eye, measured with the Freiburg Visual Acuity Test. Observers were told that they could quit the experiment at any time they wished. Five observers (two females) performed the experiment.

Apparatus and stimuli

Stimuli were presented on a HP-1332A XY-display equipped with a P11 phosphor and controlled by a PC via a custom-made 16-bit DA interface. Background luminance of the screen was below 1 cd/m^2. Luminance of stimuli was 80 cd/m^2. Luminance measurements were performed using a Minolta Luminance meter LS-100. The experimental room was dimly illuminated (0.5 lx). Viewing distance was 75 cm.

We determined vernier offset discrimination thresholds for different flanker configurations. The vernier target consisted of two lines that were randomly offset either to the left or right. Observers indicated the offset direction. Stimulus consisted of two vertical 40’ (arcmin) long lines separated by a vertical gap of 4’ and presented at an eccentricity of 5° to the right of a fixation cross (6’ diameter). Eccentricity refers to the center of the target location. Flanker configurations were centered on the vernier stimulus and were symmetrical in the horizontal dimension. Observers were presented two flanker configurations. In the lines configuration, the vernier was flanked by two vertical lines (84’) at 40’ from the vernier. In the cuboids configuration, perspective cuboids were presented to the left and to the right of the vernier (width = 58’, angle of oblique lines = 135◦, length = 23.33’). Cuboids contained the lines from the Lines condition as their centermost edge.

Procedure

Observers were instructed to fixate a fixation cross during the trial. After each response, the screen remained blank for a maximum period of 3 s during which the observer was required to make a response on vernier offset discrimination by pressing one of two push buttons. The screen was blank for 500 ms between response and the next trial.

An adaptive staircase procedure (PEST; 67) was used to determine the vernier offset for which observers reached 75% correct responses. Thresholds were determined after fitting a cumulative Gaussian to the data using probit and likelihood analyses. In order to avoid large vernier offsets, we restricted the PEST procedure to not exceed 33.3’ i.e. twice the starting value of 16.66’. Each condition was presented in separate blocks of 80 trials. All conditions were measured twice (i.e., 160 trials) and randomized individually for each observer. To compensate for possible learning effects, the order of conditions was reversed after each condition had been measured once. Auditory feedback was provided after incorrect or omitted responses.

Modelling:

To model the results of experiment 2, we trained CapsNets to solve a vernier offset discrimination task and classify verniers, cuboids, scrambled cuboids and lines. The loss function included the shape type loss, the vernier offset loss and the reconstruction loss from experiment 1 (Eqs. 2, 3, and 5):

$$L_{total} = \alpha_{shape\ type} L_{shape\ type} + \alpha_{vernier\ offset} L_{vernier\ offset} + \alpha_{reconstruction} L_{reconstruction} \quad (7)$$

The training dataset included vernier stimuli and one of three different shape types (lines, cuboids, scrambled cuboids; Figure 8a). The scrambled cuboids were included to make the task harder, and to prevent the network from classifying cuboids simply based on the number of pixels in the image. Similarly, the lines were randomly presented in a group of 2, 4, 6 or 8 to prevent the network from classifying cuboids simply based on the number of pixels in the image. Both, cuboids and shuffled cuboids were always presented in groups of two facing each another. The distance between the shapes was varied randomly between one to six pixels. As before, (un)crowding stimuli were never presented during training. Therefore, the network could not trivially learn when to (un)crowd by over-fitting on the training dataset.

After training, we tested the vernier discrimination performance for (un)crowding stimuli (Figure 8b) while varying the number of recurrent routing by agreement iterations from 1 to 8 iterations. We trained 50 networks with the same hyperparameters and averaged their performance. Some networks were excluded because vernier discrimination performance for *both* line and cuboid flankers was at ceiling (>=95%) or floor (<=55%). Qualitatively, these exclusions do not change our results. However, the fact that a few networks are at floor or ceiling is misleading.

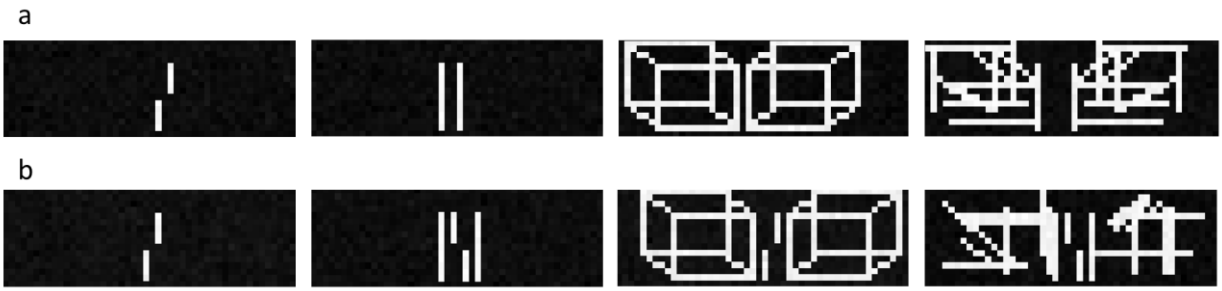


Figure 8: (a) Training stimuli: Verniers were always presented alone. Lines were presented randomly in groups of 2, 4, 6, or 8. Cuboids and scrambled cuboids were presented in groups of two facing each other. The distance between the shapes varied randomly between 1-6 pixels. **(b) Testing stimuli:** In the vernier-alone condition (*left*), we expected the network to perform well on the vernier discrimination task. In the flanker condition, we expected crowding for all chosen shape types. However, we expect that discrimination performance for the cuboid flankers recovers with increasing routing iterations.

Acknowledgements

Adrien Doerig was supported by the Swiss National Science Foundation grant n.176153 “Basics of visual processing: from elements to figures”.

Bibliography

1. A. Doerig, *et al.*, Beyond Bouma’s window: How to explain global aspects of crowding? *PLOS Computational Biology* **15**, e1006580 (2019).

2. S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules in *Advances in Neural Information Processing Systems*, (2017), pp. 3856–3866.

3. J. J. DiCarlo, D. Zoccolan, N. C. Rust, How Does the Brain Solve Visual Object Recognition? *Neuron* **73**, 415–434 (2012).

4. D. H. Hubel, T. N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology* **160**, 106–154 (1962).

5. A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks in *Advances in Neural Information Processing Systems*, (2012), pp. 1097–1105.

6. S. A. Eslami, *et al.*, Neural scene representation and rendering. *Science* **360**, 1204–1210 (2018).

7. L. Gatys, A. S. Ecker, M. Bethge, “Texture Synthesis Using Convolutional Neural Networks” in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett, Eds. (Curran Associates, Inc., 2015), pp. 262–270.

8. T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948* (2018).

9. V. A. Lamme, P. R. Roelfsema, The distinct modes of vision offered by feedforward and recurrent processing. *Trends in neurosciences* **23**, 571–579 (2000).

10. T. C. Kietzmann, *et al.*, Recurrence is required to capture the representational dynamics of the human visual system. *Proceedings of the National Academy of Sciences*, 201905544 (2019).

11. C. M. Funke, *et al.*, Comparing the ability of humans and DNNs to recognise closed contours in cluttered images in *18th Annual Meeting of the Vision Sciences Society (VSS 2018)*, (2018), p. 213.

12. J. Su, D. V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* (2019).

13. C. Szegedy, *et al.*, Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

14. R. Geirhos, *et al.*, ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231* (2018).

15. N. Baker, H. Lu, G. Erlikhman, P. J. Kellman, Deep convolutional networks do not classify based on global object shape. *PLoS computational biology* **14**, e1006613 (2018).

16. W. Brendel, M. Bethge, Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. *arXiv preprint arXiv:1904.00760* (2019).

17. T. Kim, W. Bair, A. Pasupathy, Neural coding for shape and texture in macaque area V4. *Journal of Neuroscience* **39**, 4760–4774 (2019).

18. A. Doerig, A. Bornet, O. H. Choung, M. H. Herzog, Crowding Reveals Fundamental Differences in Local vs. Global Processing in Humans and Machines. *bioRxiv*, 744268 (2019).

19. K. Hermann, S. Kornblith, Exploring CNN Inductive Biases: Shape vs. Texture. *NeurIPS Workshop on Shared Visual Representations in Human & Machine Intelligence* (2019).

20. J. Deng, *et al.*, Imagenet: A large-scale hierarchical image database in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (Ieee, 2009), pp. 248–255.

21. S.-M. Khaligh-Razavi, N. Kriegeskorte, Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS computational biology* **10**, e1003915 (2014).

22. D. L. Yamins, *et al.*, Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences* **111**, 8619–8624 (2014).

23. B. RichardWebster, S. Anthony, W. Scheirer, Psyphy: A psychophysics driven evaluation framework for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* (2018).

24. D. M. Levi, Crowding—An essential bottleneck for object recognition: A mini-review. *Vision Research* **48**, 635–654 (2008).

25. D. Whitney, D. M. Levi, Visual crowding: a fundamental limit on conscious perception and object recognition. *Trends in Cognitive Sciences* **15**, 160–168 (2011).

26. H. Bouma, Visual interference in the parafoveal recognition of initial and final letters of words. *Vision Research* **13**, 767–782 (1973).

27. D. G. Pelli, Crowding: a cortical constraint on object recognition. *Current Opinion in Neurobiology* **18**, 445–451 (2008).

28. M. Manassi, D. Whitney, Multi-level Crowding and the Paradox of Object Recognition in Clutter. *Current Biology* **28**, R127–R133 (2018).

29. M. Manassi, S. Lonchampt, A. Clarke, M. H. Herzog, What crowding can tell us about object representations. *Journal of Vision* **16**, 35–35 (2016).

30. G. Westheimer, G. Hauske, Temporal and spatial interference with vernier acuity. *Vision research* **15**, 1137–1141 (1975).

31. D. M. Levi, S. A. Klein, A. P. Aitsebaomo, Vernier acuity, crowding and cortical magnification. *Vision research* **25**, 963–977 (1985).

32. D. Oberfeld, P. Stahn, Sequential grouping modulates the effect of non-simultaneous masking on auditory intensity resolution. *PloS one* **7**, e48054 (2012).

33. K. E. Overvliet, B. Sayim, Perceptual grouping determines haptic contextual modulation. *Vision Research* **126**, 52–58 (2016).

34. T. P. Saarela, B. Sayim, G. Westheimer, M. H. Herzog, Global stimulus configuration modulates crowding. *Journal of Vision* **9**, 5–5 (2009).

35. M. H. Herzog, M. Fahle, Effects of grouping in contextual modulation. *Nature* **415**, 433 (2002).

36. B. Sayim, G. Westheimer, M. H. Herzog, Gestalt factors modulate basic spatial vision. *Psychological Science* **21**, 641–644 (2010).

37. T. P. Saarela, G. Westheimer, M. H. Herzog, The effect of spacing regularity on visual crowding. *Journal of Vision* **10**, 17–17 (2010).

38. M. Manassi, B. Sayim, M. H. Herzog, Grouping, pooling, and when bigger is better in visual crowding. *Journal of Vision* **12**, 13–13 (2012).

39. M. V. Pachai, A. C. Doerig, M. H. Herzog, How best to unify crowding? *Current Biology* **26**, R352–R353 (2016).

40. M. Manassi, B. Sayim, M. H. Herzog, When crowding of crowding leads to uncrowding. *Journal of Vision* **13**, 10–10 (2013).

41. J. M. Wallace, M. K. Chiu, A. S. Nandy, B. S. Tjan, Crowding during restricted and free viewing. *Vision Research* **84**, 50–59 (2013).

42. S. P. Tripathy, P. Cavanagh, H. E. Bedell, Large crowding zones in peripheral vision for briefly presented stimuli. *Journal of Vision* **14**, 11–11 (2014).

43. E. A. Styles, D. A. Allport, Perceptual integration of identity, location and colour. *Psychological Research* **48**, 189–200 (1986).

44. Z. Li, Visual segmentation by contextual influences via intra-cortical interactions in the primary visual cortex. *Network: computation in neural systems* **10**, 187–212 (1999).

45. L. Parkes, J. Lund, A. Angelucci, J. A. Solomon, M. Morgan, Compulsory averaging of crowded orientation signals in human vision. *Nature neuroscience* **4**, 739 (2001).

46. D. G. Pelli, M. Palomares, N. J. Majaj, Crowding is unlike ordinary masking: Distinguishing feature integration from detection. *Journal of Vision* **4**, 12–12 (2004).

47. R. Rosenholtz, D. Yu, S. Keshvari, Challenges to pooling models of crowding: Implications for visual mechanisms. *Journal of vision* **19** (2019).

48. S. Thorpe, D. Fize, C. Marlot, Speed of processing in the human visual system. *nature* **381**, 520 (1996).

49. T. C. Kietzmann, P. McClure, N. Kriegeskorte, Deep neural networks in computational neuroscience. *bioRxiv*, 133504 (2018).

50. J. Kim, D. Linsley, K. Thakkar, T. Serre, Disentangling neural mechanisms for perceptual grouping. *arXiv preprint arXiv:1906.01558* (2019).

51. H. Tang, *et al.*, Recurrent computations for visual pattern completion. *Proceedings of the National Academy of Sciences* **115**, 8835–8840 (2018).

52. C. J. Spoerer, T. C. Kietzmann, N. Kriegeskorte, Recurrent networks can recycle neural resources to flexibly trade speed for accuracy in visual recognition. *bioRxiv*, 677237 (2019).

53. D. Linsley, J. Kim, T. Serre, Sample-efficient image segmentation through recurrence. *arXiv:1811.11356 [cs]* (2018) (June 27, 2019).

54. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators. *Neural networks* **2**, 359–366 (1989).

55. N. Seijdel, N. Tsakmakidis, E. H. de Haan, S. M. Bohte, H. S. Scholte, Depth in convolutional neural networks solves scene segmentation. *bioRxiv* (2019).

56. T. Serre, Deep learning: the good, the bad, and the ugly. *Annual Review of Vision Science* **5**, 399–426 (2019).

57. Y. Qin, *et al.*, Detecting and diagnosing adversarial images with class-conditional capsule reconstructions. *arXiv preprint arXiv:1907.02957* (2019).

58. Y. Qin, N. Frosst, C. Raffel, G. Cottrell, G. Hinton, Deflecting Adversarial Attacks. *arXiv preprint arXiv:2002.07405* (2020).

59. G. E. Hinton, S. Sabour, N. Frosst, Matrix capsules with EM routing (2018).

60. P. Afshar, A. Mohammadi, K. N. Plataniotis, Brain tumor type classification via capsule networks in *2018 25th IEEE International Conference on Image Processing (ICIP)*, (IEEE, 2018), pp. 3129–3133.

61. C. J. Spoerer, P. McClure, N. Kriegeskorte, Recurrent convolutional neural networks: a better model of biological object recognition. *Frontiers in psychology* **8**, 1551 (2017).

62. K. Kar, J. Kubilius, K. Schmidt, E. B. Issa, J. J. DiCarlo, Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nature neuroscience* **22**, 974 (2019).

63. G. Francis, M. Manassi, M. H. Herzog, Neural dynamics of grouping and segmentation explain properties of visual crowding. *Psychological review* **124**, 483 (2017).

64. O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (Springer, 2015), pp. 234–241.

65. R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, K. He, *Detectron* (2018).

66. I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

67. M. Taylor, C. D. Creelman, PEST: Efficient estimates on probability functions. *The Journal of the Acoustical Society of America* **41**, 782–787 (1967).

Supplementary Material

Network hyperparameters

The following table summarizes all CapsNet hyperparameters used for the results in the main text, as well as in the supplementary material. Qualitatively, our results are robust against changes in all these network hyperparameters. This diversity highlights that our results are not the outcome of one particular set of overfitted hyperparameters. Rather, it suggests that the grouping and segmentation mechanisms in CapsNets can naturally explain the occurrence of crowding and uncrowding in our datasets.

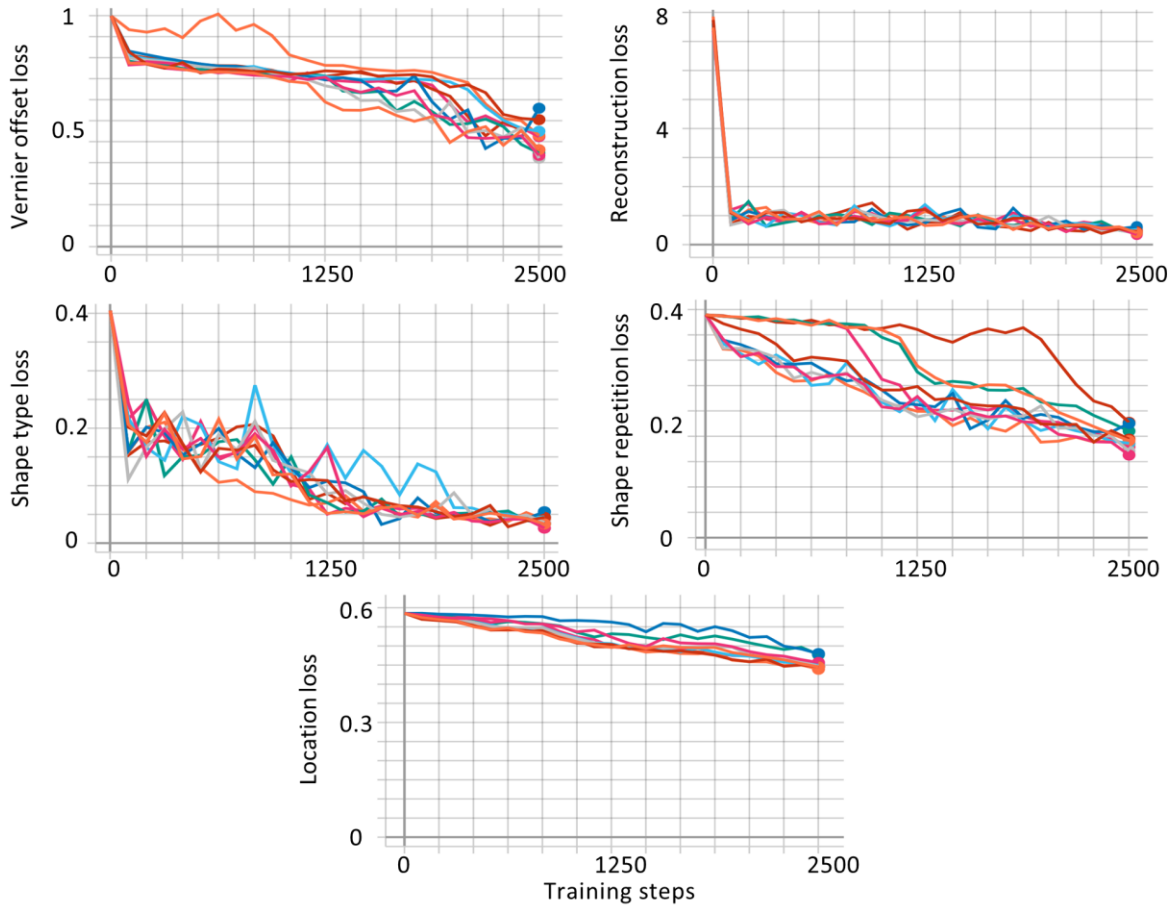
	Exp 1 Main text	Exp 2 Main text	Exp 1 Supp 1a	Exp 1 Supp 1b	Exp 1 Supp 1c	Exp 1 Supp 1d	Exp 2 Supp 3a	Exp 2 Supp 3b
Input image size	20 x 72	16 x 48	20 x 72	16 x 72	20 x 72	30 x 72	16 x 48	16 x 48
Shape size	14x14	14x11x6	14x14	14x14	14x14	14x14	14x11x6	14x11x6
Number of shape types	7	4	7	4	4	4	4	4
Shape repetitions	[1, 3, 5]	2	[1, 3, 5]	[1, 3, 5]	[1, 3, 5]	[1, 3, 5]	2	2
Number convolutional layers	3	3	3	3	3	3	3	3
Kernel sizes	5, 5, 6	4, 4, 5	5, 5, 6	5, 5, 6	5, 5, 6	5, 5, 4	4, 4, 5	4, 4, 5
Strides	1, 1, 2	1, 1, 2	1, 1, 2	1, 1, 2	1, 1, 2	1, 2, 2	1, 1, 2	1, 1, 2
Dropout used?			x	x	x	x		
Primary capsule types	7	4	7	4	4	8	4	4
Total number primary capsules	840	228	840	240	480	480	228	228
Primary capsule dimensions	2	1	2	1	1	1	1	1
Secondary capsule types	7	4	7	4	4	4	4	4
Secondary capsule dimensions	8	7	10	4	4	12	3	5
Routing iterations	3	8	3	4	4	4	8	8
Networks trained	10	50	10	20	20	20	50	50
Networks after cleaning	10	48	10	19	19	20	44	35
Gaussian training noise	[0.02, 0.04]	[0.02, 0.04]	[0.02, 0.04]	[0.0, 0.02]	[0.0, 0.0]	[0.0, 0.0]	[0.0, 0.02]	[0.0, 0.02]
Contrast adjustment delta	[0.6, 1.2]	[0.9, 1.1]	[0.6, 1.2]	[0.6, 1.2]	[0.6, 1.2]	[0.8, 1.1]	[0.9, 1.1]	[0.9, 1.1]
Gaussian test noise	[0.04, 0.06]	[0.14, 0.16]	[0.04, 0.06]	[0.1, 0.12]	[0.0, 0.02]	[0.01, 0.01]	[0.2, 0.22]	[0.06, 0.08]
Batch size	48	48	48	48	48	48	48	48
Learning rate	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
Learning rate decay steps	250	350	250	300	200	250	500	300
Training steps	2500	3500	5000	3000	2000	3000	4500	3000
Vernier offset decoder used?	x	x	x	x	x	x	x	x
Reconstruction decoder used?	x	x	x				x	x
Shape repetition decoder used?	x		x	x	x	x		
Location decoder used?	x		x					

Supp. Table 1: Network hyperparameters. The rows show all hyperparameters and the columns show their selected values for the different networks. Parameters for the main text results are highlighted. The number of convolutional layers includes the primary capsule layer. The total number of primary capsules is equal to the output size of the last convolutional layer multiplied with the number of primary capsule types.

Experiment 1

Loss evolution during training

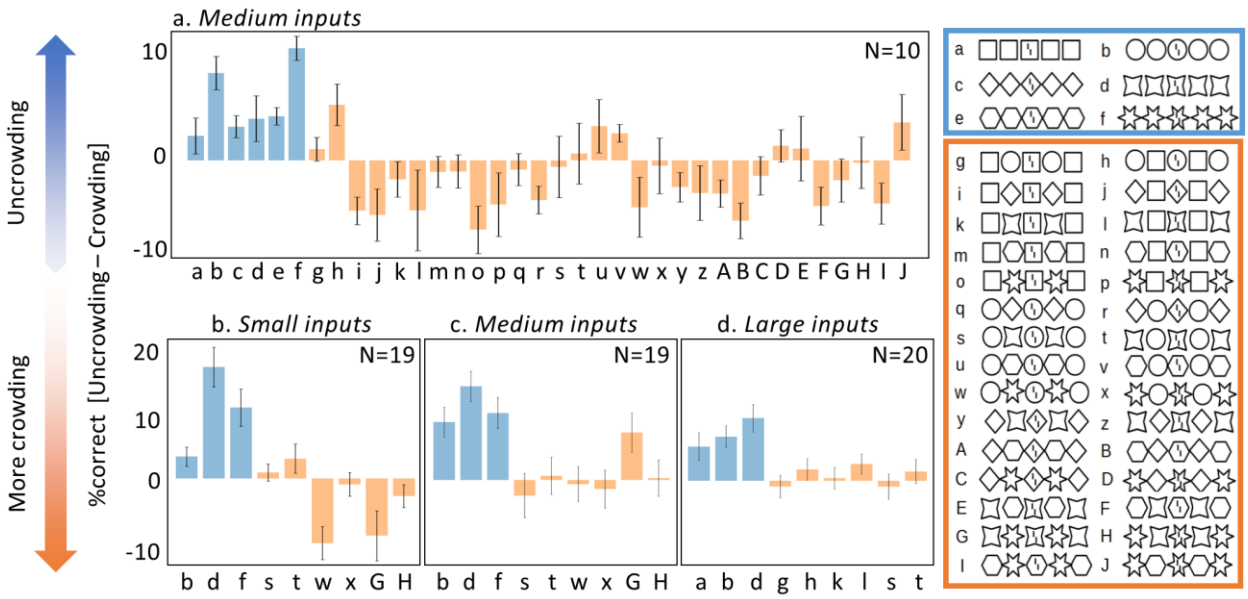
To exemplify the evolution of the loss, the following figure shows the progress of each loss term during training for each of the N=10 networks of the main results. The loss terms were scaled to ensure that none of them was dominant during training. All losses were minimized simultaneously.



Supplementary Figure 1: Loss evolution during training. The x-axis shows the number of training steps and the y-axes shows the absolute loss values for each loss term. Each color represents one of the N=10 networks that was trained for the main results of experiment 1. All losses decreased during training.

Results are robust against hyperparameters changes

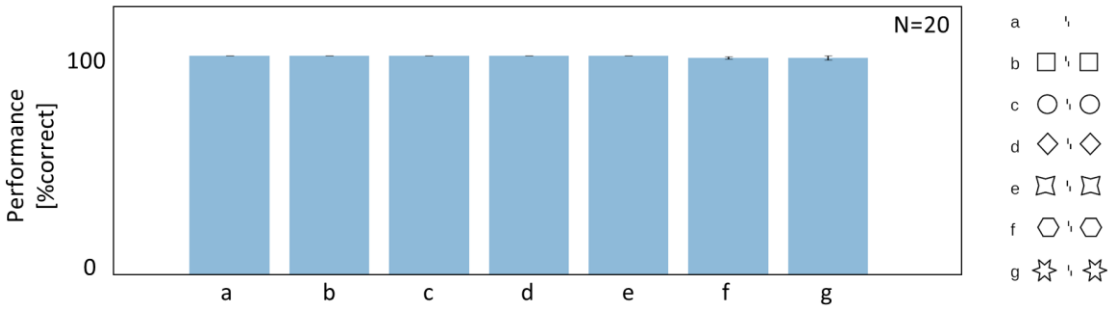
To show that the reproduction of (un)crowding is a general trait of CapsNets, we trained more networks with different hyperparameter sets. The robustness of these results stresses the importance of the underlying network architecture for (un)crowding. Few networks were excluded because vernier discrimination performance in the vernier-alone condition was floored (<=55%).



Supplementary Figure 2: Results are robust against hyperparameter changes. The x-axes show the configurations that were used and the y-axes show the vernier discrimination performance as %correct for the whole configuration minus %correct for the central flanker alone. Error bars indicate the standard error between trained networks. Crowding and uncrowding occur qualitatively similar in all networks with different hyperparameters using (b) small input images of size 16x72 pixels, (a, c) medium input images of size 20x72 pixels, and (d) larger input images of size 30x72 pixels.

Performance deterioration is due to crowding

To show that crowding occurred in our networks due to the proximity of the vernier target and the flankers, we ran a control experiment in which the vernier was presented outside the flanking shapes. Vernier discrimination accuracy did not drop in this case. This suggests that the performance drop can be ascribed to crowding and not some other peculiarity in the networks.

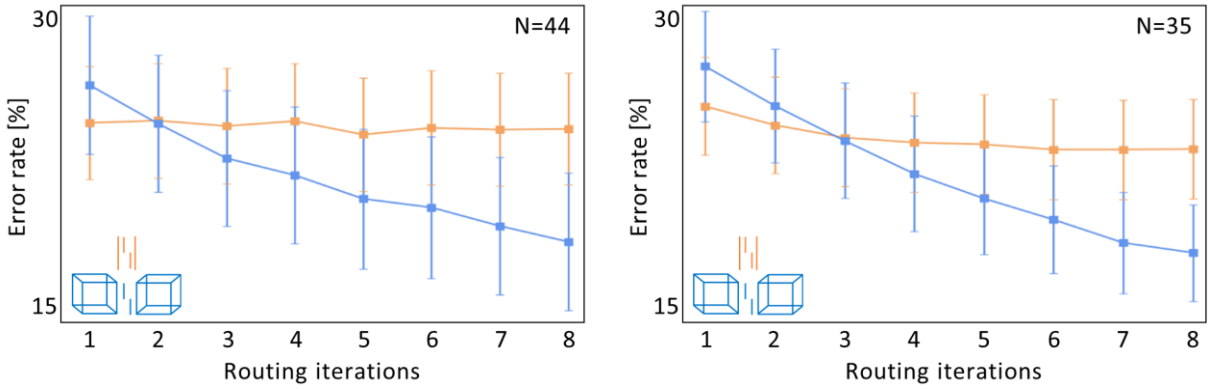


Supplementary Figure 3: Performance deteriorates due to crowding. The x-axis shows the configurations that were used and the y-axis shows the vernier discrimination performance. Error bars indicate the standard error between trained networks. Vernier discrimination accuracy does not decrease when the vernier is presented outside the flanking shapes (configurations b-g) compared to the vernier alone condition (configuration a).

Experiment 2

Results are robust against hyperparameters changes

As for experiment 1, we ran several CapsNets with different hyperparameter sets, and show that our results are robust with respect to these changes. In both of the following examples, the improvement in vernier discrimination performance with an increasing number of recurrent routing iterations was greater in the cuboids condition than in the lines condition (left: p=0.0008, right: p=0.003, 2-tailed two-sample t-tests). This stresses the importance of the underlying network architecture for reproducing the temporal dynamics of (un)crowding. Some networks were excluded because vernier discrimination performance for line and cuboid flankers was at ceiling (>=95%) or floor (<=55%).



Supplementary Figure 4: Results are robust against hyperparameter changes. The x-axis shows the number of recurrent routing iterations and the y-axis shows the error rates for the vernier discrimination performance (i.e., lower values indicate better performance). Error bars indicate the standard error between trained networks. As for the main results, performance increases significantly more with routing iterations in the cuboids condition (blue) as compared to the lines condition (orange), reproducing human behavior.