Cross-Domain and Within-Domain Synaptic Maintenance for Autonomous Development of Visual Areas

Qian Guo, Xiaofeng Wu, and Juyang Weng

***Abstract*—Where-What Networks (WWNs) is a series of developmental networks for the recognition and attention of complex visual scenes. One of the most critical challenges of autonomous development is task non-specificity, namely, the network is meant to learn a variety of open-ended task skills without pre-defined tasks. Then how does a brain-like network develop skills for object relation that can generalize using implicit symbol-like rules? A preliminary scheme of uniform synaptic maintenance, which works across a neuron's sensory and motor domains, has been proposed in our WWN-9. In the new work here, we show that cross-domain and within-domain synaptic maintenance gains superior generalization than using the uniform synaptic maintenance scheme. This generalization enables the WWN to automatically discover symbol-like but implicit rules — detecting object groups from new combinations of object locations that were never observed. By "symbol-like but implicit rules", we mean that the development program has no symbols and explicit rules, but symbol-like concepts (location, type) and implicit rule (two specific type objects must present concurrently — group) emerge as the firing patterns of the motor area and are used by the control. Moreover, the process of synaptic maintenance corresponds to the genesis (and adaptation) of cell connections and our model autonomously develops the Y area into two subarea, early area and later area, in charge of pattern recognition and symbolic reasoning respectively.**

I. Introduction

A critical question for neuroscience and artificial intelligence is to understand how visual cortex recognizes objects, because humans outperform the best machine vision system on many aspects [10]. It's natural to think of building a brain-like network to improve machine vision performance. Visual processing cortex is traditionally modeled as a distributed hierarchy of visual areas and of somatosensory / motor areas based on a total of 305 connections among 32 visual and visual-association areas [2].

With the advances of the studies on visual cortex in physiology and neuroscience, several biological-inspired network models have been proposed. The earliest work in this area can be traced back to 1959, the structure of cat striate cortex has been studied by Hubel and Wiesel [6]. In 1962, they proposed a layered network model [7] based on the visual cortex mechanism. In 1980, Fukushima proposed self-organizing, multi-layered structure, feedforward neural network model Neocognitron [3]. Multi-layer structure of the network gradually increases the complexity of the features and form position invariance. Compared to the manual design of layer in Neocognition, Cresceptron by Weng [14] can perceive the outside world information and generate neuron layers independently and incrementally. However, it does not provide an effective mechanism to learn about other types of invariance, such as scale invariance, illumination invariance and so on.

Applied some biological mechanisms such as lateral competition, edge features extraction by Gabor filters and feature combination, the feed-forward network HMAX proposed by Poggio [10] in 1999 simulates the ventral pathway in primate vision system to do object recognition with the loss of object location. Note that max-pooling techniques were used in Cresceptron, although terms such as max-pooling and HMAX were used later. The deep neural networks in recent years [5] adopted the hierarchy like cortex in both architecture (multiply hidden layers) and processing flow (deep learning algorithm) and acquired much better results than traditional shallow networks. But it is difficult for these models to utilize top-down attention and the design rules of network structure are empirical rather than theoretical grounded.

Where-What Networks (WWNs) [8] first introduced in 2008 is a biologically plausible developmental model designed to integrate the object recognition and attention namely, "what" and "where" information in the ventral stream and dorsal stream respectively. Therefore, multiple concepts (e.g., type, location, scale) can be learned concurrently in the network. That is to say, the feature representation, object classification and detection can be accomplished in such a single network. Different from other models, the processing modules are organized in an emergent network instead of a rigid cascade, for both computational principles (statistics) and biological plausibility [2]. It uses both feedforward (bottom-up) and feedback (top-down) connections so that the network can work in different attention modes (e.g. location given, type given or free viewing) according to top-down instructions. With the lateral inhibitions (competition), only the best matching neurons update so that the network can learn new objects quickly without impacting previously learned useful memory.

So far, a series of biological inspired mechanisms based on the method of autonomous development have been introduced in WWNs. The mechanism of synaptic maintenance

Qian Guo and Xiaofeng Wu (Corresponding author), are with Department of Electronic Engineering, Fudan University, Shanghai, 200433, China, (email: {13210720028, xiaofengwu} @fudan.edu.cn); Juyang Weng was a visiting professor at the School of Computer Science, Fudan University, Shanghai, 200433, China 2003-2014 and is a professor at Department of Computer Science and Engineering, Cognitive Science Program, and Neuroscience Program, Michigan State University, East lansing, Michigan, 48824, USA, (email:weng@cse.msu.edu); This work was supported by Changjiang Scholar Fund of China to JW.

was first proposed to automatically adapt the shape of receptive field of a neuron. The result shows that the adaptive receptive fields can reduce the interference from leaked-in background pixels which have larger fluctuations than the foreground pixels [12]. Then the synaptic maintenance was extended from sensory domain to multiple domains in WWN-9 and was named cross-domain synaptic maintenance. It refined the earlier connections to enable each neuron to autonomously find its role. However, it was a preliminary work and the network performance wasn't improved when cross-domain synaptic maintenance was applied [4].

In the new work here, we revise the algorithms of synaptic maintenance based on WWN-9 and show our model can recognize and locate the group in the complex backgrounds where the layout of members was never observed. The synaptic maintenance consists of two stages: within-domain and cross-domain instead of the previous uniform scheme. The within-domain synaptic maintenance compares the deviations of input signals and do trimming cell-wise. It tends to extend the input synapses with small noises and contract the synapses with big noises. The cross-domain trimming compares the mean deviation of a whole domain or subdomain to do trimming domain-wise. It adjusts the contribution of each domain for each neuron by trimming or even completely cutting it off. We show that cross-domain and within-domain synaptic maintenance gains superior generalization than using a uniform synaptic maintenance scheme. It is known that thalamus — the source of signals for cerebral cortex — are important for specifying the patterns of brain areas in the cerebral cortex [1]. So we further model the autonomous development of early area and later area in the cortex through cross-domain and within-domain synaptic maintenance.

The remainder of this paper is organized as follows. Section II introduces the architecture and operation of WWN-9. Section III presents some important concepts and algorithms in the network. Experimental results are reported in Section IV. Section V gives the concluding remarks.

II. Where-What Networks Overview

In this section, the network structure and the learning algorithm LCA used in WWNs are briefly described. The reader is recommended to browse our prior work first because of the page limit of each paper.

A. Network Structure

The architecture of WWN-9 (shown in Fig. 1) consists of three areas, X area (retina / sensors), Y area (cortex / control center) and Z area (motor ends / effectors). X acts as the retina, which perceives the inputs and sends signals to Y. Z is made up of subareas like Location Motor (LM) and Type Motor (TM) which serve as both input and output. The arm muscles determine where the object is and we call them Location Motor (LM). The muscles of vocal tract producing name of the object and we call them Type Motor (TM). During supervised training, the external world provides top-down input (e.g. location and name of object) to network. Otherwise, Z gives the output to drive effectors which act on the real world. Functionally, Z is used as the hub for emergent concepts (e.g., location, scale and type), abstraction (many forms mapped to one equivalent state), and reasoning (as goal-dependent emergent action). In the structure of WWN-9, besides the LM and TM, a new sub-area called Group Motor (GM) is added, which corresponds to group relation. It includes the concepts of object pairs, such as AB, AC and CD. Y area is like a "bridge" linking both X and Z as its two "banks" through 2-way connections. It is worth to note that Y area is completely inside the closed skull, which is off limit to the teachers in the external environments.

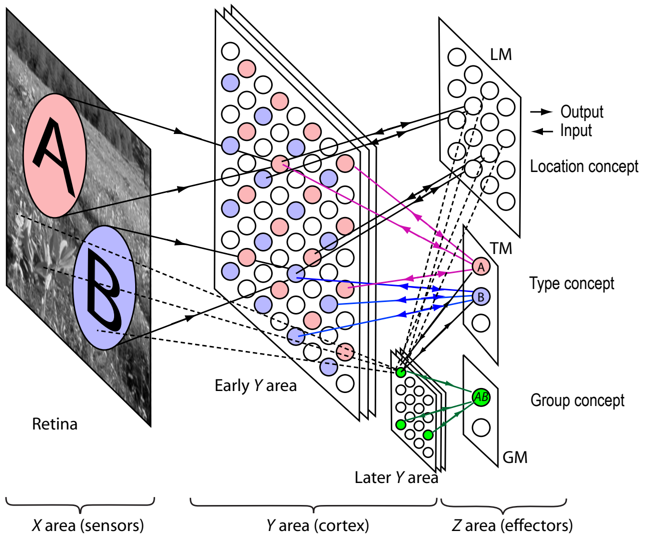


Fig. 1. The architecture of WWN-9 using cross-domain synaptic maintenance to autonomously specify early and later areas. The dashed lines means that the connection between two areas are cut off. One two-way arrow means two one-way arrows because of the limited space. Note that the receptive fields of later Y area are sparse and non-local.

Different with the static structure of previous WWNs, the morphology of WWN-9 structure is expected to be changed with development automatically. The pattern of visual areas are specified that early Y mainly handles inputs from X while later Y mainly processes signals from Z (TM, LM and GM).

B. Lobe Component Analysis Algorithm

There are many learning algorithms that aim to determine the weights for a set of neurons using observations (data). We use lobe component analysis (LCA) [15] which has the dual optimality ("best" and "fastest"). As this algorithm has been applied by previous versions of WWN, only the most important parts of it are described as follows.

1) Pre-response Computation.
Compute the pre-response

$$r(\mathbf{v}, \mathbf{p}) = \dot{\mathbf{v}} \cdot \dot{\mathbf{p}}, \tag{1}$$

where $\dot{\mathbf{v}}$ is the normalized weight vector, and $\dot{\mathbf{p}}$ is the normalized input vector. The inner product measures the degree of match between these two directions of unit vector $\dot{\mathbf{v}}$ and $\dot{\mathbf{p}}$, because $r(\mathbf{v}, \mathbf{p}) = \cos(\theta)$ where θ is the angle between $\dot{\mathbf{v}}$ and $\dot{\mathbf{p}}$.

2) Top-k Competition

The response r'_q after top-k competition is

$$r'_q = \begin{cases} (r_q - r_{k+1})/(r_1 - r_{k+1}) & \text{if } 1 \leq q \leq k \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where r_1, r_q and r_{k+1} denote the first, qth and $(k+1)$th neuron's pre-response respectively after being sorted in descending order.

3) Hebbian-like learning

The synaptic weights update using amnesic average. Different from incremental average, it gradually forgets the old information for a non-stationary process.

$$\mathbf{v}_j(n) = w_1(n)\mathbf{v}_j(n-1) + w_2(n)r'_j(t)\dot{\mathbf{p}}_j(t), \tag{3}$$

where $r'_j(t)$ is the response of the neuron j after top-k competition, n is neuron's age, namely the firing times of the neuron, $\mathbf{v}_j(n)$ is the synaptic weights vector of the neuron, $\dot{\mathbf{p}}_j(t)$ is the input patch perceived by the neuron, w_1 and w_2 are two parameters representing retention rate and learning rate with $w_1 + w_2 \equiv 1$.

C. Two Types of Receptive Fields

The Y neurons in different subareas have different types of receptive fields at initial stage. Neurons in early Y area whose synapses with X are arranged regularly and tightly in space have fixed and intensive receptive fields, thus they can detect stable features from input images. The synaptic connections are shown by the solid arrows between early Y area and retina in Fig. 1.

However, neurons in later Y area whose synapses with X are arranged loosely and randomly in space have non-local and sparse receptive fields, thus their input vectors can not be extracted to stable features for recognition. The synaptic connections are shown by the dashed lines between later Y area and retina in Fig. 1, the dashed lines in which means the synapses are cut off by domain trimming.

To sum up, the two kinds of receptive fields provide neuronal signals of different statistical properties to enable the network to specify the patterns of visual areas.

III. Concepts and Theory

It is known that the biological structures of the brain develop soon after the conception, throughout much of the prenatal period and continues through the postnatal period. Cells in a brain include (A) neurons that compute and (B) glia that play supportive roles (blood supply) for neurons. Three major stages of development have been identified [9], [11]: (1) cell proliferation in which new cells are generated from mitosis (cell splits); (2) cell migration in which cells move according to morphogens that are substances released by other cells including neurons that fire; and (3) cell differentiation where cells change into highly differentiated (i.e., highly specialized) cell types depending on the cells activities (e.g., neuronal firing). The stage (3) determines the precise wiring among neurons. However, morphogens in stage (2) only roughly guide the migration of cells. Precise wiring between any two neurons includes two decisions, connect or not, and if connect, how to connect. Both parts heavily depend on the activities of both neurons, such as where each is located in the brain and their firing patterns. In our model, two neurons connect or not depends on at least two processes: (a) synaptic maintenance which determines whether a synapse is generated (or retracted/cut) between the presynaptic (sending) neuron and postsynaptic (receiving) neuron and (b) Hebbian learning which determines how strong the conductance (weight) of a synapse is according to the firing activities of the post-synaptic neuron and post-synaptic neuron. We do not simulate stages (1) and (2) because it seems to be more computationally efficient for the human programmer to handcraft the number of cells and the extent to which neurons can receive inputs from. However, a human programmer is unlikely to handcraft stage (3) well. In this paper, we concentrate on the process (a) since process (b) applied in WWNs has been discussed in the previous work. Synaptic maintenance in organisms is highly complex. Computationally, we use what is called synaptogenic factor in biology as a multiplicative but highly dynamic number between two potentially linkable neurons.

Compared with a uniform synaptic maintenance scheme used in [4], we proposed a two stage scheme including cross-domain and within-domain synaptic maintenance. Based on the idea of neuromodulation, we proposed the domain trimming which is critical for the development of connectivity patterns of visual areas.

A. Within-Domain Synaptic Maintenance

The process of synaptic maintenance is the modulation of synaptogenic factors f which is determined by the standard deviations σ of synaptic inputs p. The synapse i updates incrementally using amnesic average, which enable the standard deviation to be plastic continuously.

The expression of standard deviations $\sigma_i(n)$ and its derivation were given in [4], so it won't be covered again here. The expected synaptic deviation among all the synapses of a neuron is defined by:

$$\bar{\sigma}_j(n) = \frac{1}{d(n)} \sum_{i=1}^{d(n)} \sigma_i(n). \tag{4}$$

The relative ratio is defined as:

$$r_i(n) = \frac{\sigma_i(n)}{\bar{\sigma}(n)}. \tag{5}$$

We would like to retract synapse i if $\sigma_i(n)$ is large. However, we do not want a hard threshold that may cause a synapse to be retracted and extracted repeatedly. For this reason, we introduce a smooth synaptogenic factor $f(r)$ to determine the percentage of trimming.

The synaptogenic factor that uses three linear segments is

$$f(\sigma_{ij}, \bar{\sigma}) = \begin{cases} 1 & \text{if } \sigma_{ij}/\bar{\sigma} < \beta_s \\ (\beta_b - r_{ij})/(\beta_b - \beta_s) & \text{if } \beta_s \leq \sigma_{ij}/\bar{\sigma} \leq \beta_b \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

We tried $\beta_s = 1.0$ and $\beta_b = 1.5$. Namely, if all the synapses matched equally well, $r_i(n) = 1$ for all i. Then $f(r_i(n)) = 1$, to keep them all. If the degree of synaptic matches have discrepancy, then those synapses with $r_i(n) \leq \beta_s$ are still fully operational. Those with $\beta_s < r_i(n) < \beta_b$ are partially cut, and those with $\beta_b < r_i(n)$ are completely cut.

We use the synaptogenic factor as a multiplicative factor to model the retraction and extraction of synapse. The trimmed weight vector $\mathbf{v}' = (v'_1, v'_2, ..., v'_d)$ is defined by an element-wise multiplication.

$$v'_i = f_i v_i, i = 1, 2, ..., d. \tag{7}$$

Similarly, we can get the trimmed input vector $\mathbf{p}' = (p'_1, p'_2, ..., p'_d)$.

B. Cross-Domain Synaptic Maintenance: Domain Trimming

In the WWN model, each neuron in Y has three domains of input: bottom-up $\mathbf{b}$, lateral $\mathbf{l}$ and top-down $\mathbf{t}$. A domain may have several sub-domains. For example, the top-down domain $\mathbf{t}$ is Z area which has three sub-domains, LM, TM and GM. The bottom-up domain $\mathbf{b}$ is X area and lateral domain $\mathbf{l}$ is Y area itself.

The above within-domain synaptic maintenance is the trimming of connection between cells. We will further explore the synaptic maintenance in all the input domains and the trimming in domain level, that is cross-domain synaptic maintenance. Although the input from different domains are pixel intensity or motor state (e.g. firing or not), each input vector has already normalized into the range $[0, 1]$ where 0 means not firing and 1 means firing (or brightest pixel). Therefore, the standard deviation σ of synapses from different domains can be compared.

The cross-domain stage is based on the within-domain algorithms. We define the expected synaptic deviation across all the domains as follows.

$$\bar{\sigma}(n) = \frac{1}{s(n)} \sum_{j=1}^{s(n)} \bar{\sigma}_j(n), \tag{8}$$

where the $\bar{\sigma}_j(n)$ is same as the expected deviation of one domain in Eq. (4). $s(n)$ is the number of domains (or subdomains).

A domain is completely cut off from the cell if the ratio of its expected synaptic value to the cross-domain expected synaptic value, measured by $\bar{\sigma}_j / \bar{\sigma}$ is bigger than β_b in Eq. (6).

The domain trimming is scaling the vector by the cross-domain synaptogenic factor. The trimmed weight vector $\mathbf{v}''_j$ is based on the $\mathbf{v}'_j$ computed in the first stage.

$$\mathbf{v}''_j = f_j \mathbf{v}'_j, \tag{9}$$

where $\mathbf{v}'_j$ is the weight vector from domain j and f_j is the synaptogenic factor for domain j.

Then this trimming process is done for the input vector similarly and the calculation of pre-response should be modified with the trimmed vectors. Therefore the cross-domain synaptogenic factor dynamically determines whether the corresponding domain provides a full supply, no supply, or in between. Domain trimming deals with global self-wiring, such as which areas are the input domains of a neuron. One example is that the later Y area will gradually cut off the connections from the domain X area so that it takes input from the domain Z areas only. Such later Y area will perform more complex functions such as relation between objects (e.g., group).

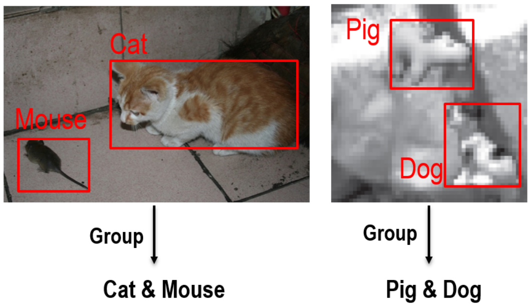


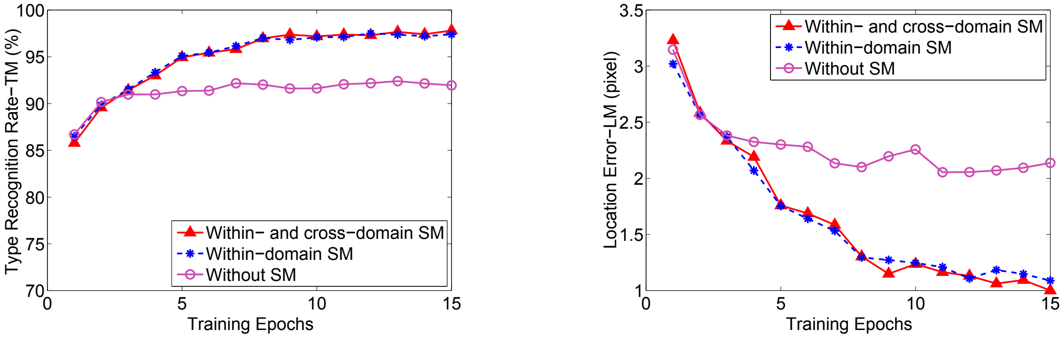
Fig. 3. The illustrative example of multi-object detection and object group recognition in the experiment.

Note that the within-domain trimming discussed earlier should be done first for each domain and the cross-domain trimming is done later. Therefore, with the cross-domain synaptic maintenance, the network can dynamically determine which domain to be cut off at the same time fine tune the connections within each domain.

C. LCA Algorithm with Synaptic Maintenance

We update the LCA algorithm introduced in II-B with synaptic maintenance to be added. The revised algorithm is described as follows:

1) Weighting: Every neuron performs first the range normalization and then the ϵ-mean normalization for each of $\mathbf{b}, \mathbf{l}$ and $\mathbf{t}$ respectively, of the input vector $\mathbf{p} = (\mathbf{b}, \mathbf{l}, \mathbf{t})$.
2) Cell trimming: Every neuron trims its inputs and weights using the within-domain synaptogenic factors (shown in Eq. (7)).
3) Re-weighting: Every neuron performs the range normalization and the ϵ-mean normalization for its trimmed input vectors and weight vectors.
4) Domain trimming: Every neuron trims its inputs and weights using the cross-domain synaptogenic factors (shown in Eq. (9)).
5) Every neuron computes the pre-action potential.
6) Conduct top-k competition for all neurons in the same domain.
7) Only the winner neurons conducts Hebbian learning to update its synaptic weight vector.
8) Only the winner neurons update σ_i and the synaptogenetic factors and advance their age.



(a) Type Recognition Rate-TM

(b) Location Error-LM

Fig. 2. Network performance variation in 15 epochs in three cases. Case 1: with both within-domain and cross-domain synaptic maintenance; case 2: only with within-domain synaptic maintenance; case 3: without any synaptic maintenance. SM is short for synaptic maintenance.

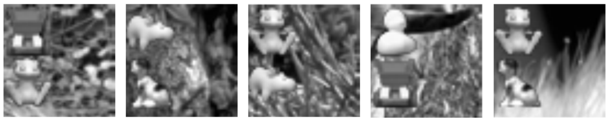


Fig. 4. Some actual input images of object group samples.

IV. Experiments and results

In previous works, we showed that WWNs can deal with attention and recognition of single foreground object in complex backgrounds [13]. However, the network's capability of multi-object detection has not been investigated. In the experiments, we show that WWN-9 can deal with multi-object detection in clutter background and based on that recognize object groups (illustrated in Fig. 3). Then the effect of cross-domain and within-domain synaptic maintenance and their implications are discussed.

A. Experiment design

Inspired by the natural images of two animals as a group, we constructed our samples using toy animals (shown in Fig. 4). In the experiment samples, the foreground objects are selected from the MSU 25-objects data set, with label of A, B, C, D and E. Totally 5 different types of objects are paired into 10 groups which are labeled as AB, AC, AD, AE, BC, BD, BE, CD, CE and DE. The possible locations of object are 30×30 (i.e., LM has 30×30 neurons).

To verify the network, we tested the network with sufficient neuron resources. At epoch 15, the recognition rate reached over 98% and location error dropped to about 0.55 pixel for single object, which we considered enough for the next stage of groups' learning. After training had progressed enough to get a good performance (let's say type recognition rate is above 95%), we saved the trained network and used it to recognize two objects in the scene simultaneously.

The experiment I proves the effect of within-domain trimming, namely improving the type recognition rate and reducing location error when recognizing multiple objects (shown in Fig. 2). The experiment II shows that the effect of cross-domain trimming is to improve the group recognition rate (shown in Fig. 5).

B. Experiment I: multi-object detection

In our model, the key point for multi-object detection is to increase the value of K in the top-k competition. Given an input image, each Y neuron will be activated to have a pre-response value. Then after the top-k competition, the response values of two winner neurons are retained while others are cleared. It's worth to note that the best matcher's neighbors are inhibited to let the neuron corresponding to another object stand out.

At each epoch, the network learns single objects with supervision of an external teacher. During the training of single objects, the teacher only provides the corresponding correct information in TM and LM so that no neuron in GM fires. A test in the free-viewing mode is performed after each training epoch. The free-viewing mode means that the network works without any teacher's supervision of type or location. The Fig. 2 shows that the recognition rate reaches 97.55% and the location error is 1.05 pixel after 15 epochs. Compared with the case without any synaptic maintenance, the type error and the location error has significantly reduced 72.82% and 53.17%, respectively, when using synaptic maintenance. Note that it's the contribution of within-domain trimming, not the cross-domain trimming. Because the within-domain trimming can reduce the interference from leaked-in background pixels.

C. Experiment II: object group recognition

In experiment II, we train the network with object group based on the saved network pre-trained by single object, since good recognition performance of type and location is the basis of learning group relation. For single object training, each single object needs to be trained at every possible position. However, for object group training, we don't do such exhaustive training for all the position combinations. In Fig. 5, the group recognition rate without domain trimming drops to 55.76%, which means that the network can not recognize groups where the member layouts have never been

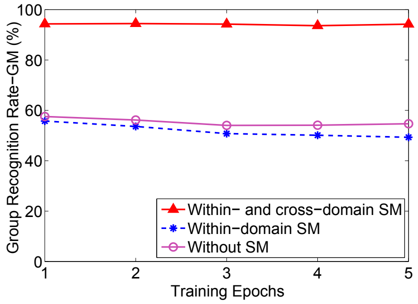


Fig. 5. Group recognition rate variation with the training of object groups. It's based on the network trained by single object in epoch 15. Case 1: with both within-domain and cross-domain synaptic maintenance, case 2: only with within-domain synaptic maintenance, case 3: without any synaptic maintenance.

observed. By contrast, the group recognition rate rises to 94.33% since the later area is location invariant with domain trimming.

By the weights visualization, we found that for later Y area the domain synaptogenic factors of X and LM are all trimmed to zeros. It means that the later Y area has learnt the symbolic-like group relation between TM and GM at the exclusion of X and LM. By cutting off connections with X and LM completely, the later Y area achieves bottom up invariance and position invariance.

D. Discussion

In the previous work [4], the object group recognition with cross-domain synaptic maintenance scheme are not improved because of the imperfection in algorithms and experiment design. The revised mechanism enables the WWN-9 to recognize object group in a part-based way and learn that two objects can be considered as a group, independent of the layout, which is an implicit rule embedded in the training samples. Moreover, our model can generalize the rule from a few samples rather than the big data.

V. Conclusions and Future Work

The proposed biologically inspired mechanism of cross-domain and within-domain synaptic maintenance enables the WWN to automatically discover symbol-like but implicit rules — detecting object groups from new combinations of object locations that were never observed. Although this mechanism has a potential to develop other symbol-like relations, our experimental work here only limits to the presence of two-object groups.

In the future work, we will study the change of network structure by analysis of visualization of the neurons in the Y area with cross-domain and within-domain synaptogenic factor. Experiments on larger and more practical dataset will be done.

References

[1] M. F. Bear, B. W. Connors, and M. A. Paradiso. *Neuroscience: Exploring the Brain*. Lippincott Williams & Wilkins, Philadelphia, 3rd edition, 2007.

[2] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1:1–47, 1991.

[3] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[4] Q. Guo, X. Wu, and J. Weng. Wwn-9: Cross-domain synaptic maintenance and its application to object groups recognition. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 716–723. IEEE, 2014.

[5] G. E. Hinton, S. Osindero, and Y-. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[6] D. H. Hubel and T. N. Wiesel. Receptive feilds of single neurons in the cat's striate cortex. *Journal of Physiology*, 148:574–591, 1959.

[7] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160(1):107–155, 1962.

[8] Z. Ji, J. Weng, and D. Prokhorov. Where-what network 1:where and what assist each other through top-down connections. In *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*, pages 61–66. IEEE, 2008.

[9] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, and A. J. Hudspeth, editors. *Principles of Neural Science*. McGraw-Hill, New York, 5th edition, 2012.

[10] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007.

[11] M. Sur and J. L. R. Rubenstein. Patterning and plasticity of the cerebral cortex. *Science*, 310:805–810, 2005.

[12] Y. Wang, X. Wu, and J. Weng. Synapse maintenance in the where-what networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2822–2829. IEEE, 2011.

[13] Y. Wang, X. Wu, and J. Weng. Skull-closed autonomous development: Wwn-6 using natural video. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.

[14] J. Weng, N. Ahuja, and T. S. Huang. Learning recognition and segmentation using the Cresceptron. *International Journal of Computer Vision*, 25(2):109–143, Nov. 1997.

[15] J. Weng and M. Luciw. Dually optimal neuronal layers: Lobe component analysis. *IEEE Trans. Autonomous Mental Development*, 1(1):68–85, 2009.