
THE DYNAMIC NET ARCHITECTURE: LEARNING ROBUST AND HOLISTIC VISUAL REPRESENTATIONS THROUGH SELF-ORGANIZING NETWORKS

Pascal J. Sager

Zurich University of Applied Sciences
University of Zurich
sage@zhaw.ch

Jan M. Deriu

Zurich University of Applied Sciences
deri@zhaw.ch

Benjamin F. Grewe

University of Zurich, ETH Zurich
bgrewe@ethz.ch

Thilo Stadelmann

Zurich University of Applied Sciences
European Centre for Living Technology (Venice, IT)
stdm@zhaw.ch

Christoph von der Malsburg

Frankfurt Institute for Advanced Studies
University of Zurich, ETH Zurich
malsburg@fias.uni-frankfurt.de

July 8, 2024

ABSTRACT

We present a novel intelligent-system architecture called “Dynamic Net Architecture” (DNA) that relies on recurrence-stabilized networks and discuss it in application to vision. Our architecture models a (cerebral cortical) area wherein elementary feature neurons encode details of visual structures, and coherent nets of such neurons model holistic object structures. By interpreting smaller or larger coherent pieces of an area network as complex features, our model encodes hierarchical feature representations essentially different than artificial neural networks (ANNs).

DNA models operate on a dynamic connectionism principle, wherein neural activations stemming from initial afferent signals undergo stabilization through a self-organizing mechanism facilitated by Hebbian plasticity alongside periodically tightening inhibition. In contrast to ANNs, which rely on feed-forward connections and backpropagation of error, we posit that this processing paradigm leads to highly robust representations, as by employing dynamic lateral connections, irrelevant details in neural activations are filtered out, freeing further processing steps from distracting noise and premature decisions.

We empirically demonstrate the viability of the DNA by composing line fragments into longer lines and show that the construction of nets representing lines remains robust even with the introduction of up to 59% noise at each spatial location. Furthermore, we demonstrate the model’s capability to reconstruct anticipated features from partially obscured inputs and that it can generalize to patterns not observed during training. In this work, we limit the DNA to one cortical area and focus on its internals while providing insights into a standalone area’s strengths and shortcomings. Additionally, we provide an outlook on how future work can implement invariant object recognition by combining multiple areas.

Keywords net fragments · neural code · early commitment · pattern recognition · computer vision · neuronal networks · neuroscience · machine learning

1 Introduction

We present the first implementation of a novel intelligent-system architecture called “Dynamic Net Architecture” (DNA). We motivate it here by contrasting it to the architectures of artificial neural networks (ANNs) that play an essential role in analyzing high-dimensional data (LeCun et al., 2015; Schmidhuber, 2015; Stadelmann et al., 2018). An ANN is a static, deeply nested function optimized with respect to a static cost function. An input to an ANN is mapped to an output by a succession of nested functions arranged in layers. Consequently, representations have the form of a hierarchy that integrates simpler low-level features (outputs of inner functions) to more complex high-level features (outputs of outer functions). The representations computed by the inner functions are independent of the outputs of outer functions. We posit that this is one of the leading causes of the lack of robustness witnessed in ANNs (Windrim et al., 2016; Li et al., 2019; Wang et al., 2019; Rusak et al., 2020; Drenkow et al., 2022) since the decisions of inner functions cannot be adjusted by plausibility checks based on outer functions. This assertion finds support in the formal explanation provided by Goodfellow et al. (2015), who argue that adversarial attacks rely on perturbations below the threshold of visibility when these are summed over a sufficiently high input dimensionality (cf. Ilyas et al. 2019). Following Marr (2010), we call this phenomenon of lacking robustness due to sequential information processing “the fallacy of early commitment,” i.e., the inability to factor the outputs of the outer functions (high-level features) into the computation of the inner functions (low-level features), so that low-level decisions are in danger of misleading high-level decisions.

With the proposed DNA, we overcome this limitation by dynamically integrating local and global features. The DNA is formulated in terms of cortical areas (two-dimensional sheets of neurons) and projections between them. Connectivity within areas is dominated by short-range lateral (“recurrent”) connections, different areas are connected bidirectionally. The neurons of a given area are elementary feature detectors. Afferent connections from a given input pattern activate a fraction of all neurons in the area. Cyclically rising inhibition then silences the majority of the initially activated neurons, leaving only those active that are supporting each other with sufficient strength through excitatory lateral connections. This mutual support can only be realized within specifically structured consistent networks, “nets” (von der Malsburg, 2018) and have to be generated by statistical learning. Nets can span the entire visual field and connect feature neurons that are spatially distant through chains of locally connected intermediary neurons. Nets are flexibly composed of “net fragments” that correspond to often-occurring feature constellations. Each position of an area contains many fragments (analogous to the code-book of image compression schemes, Taubman and Marcellin (2002)) from which the input can choose. Unlike the pieces of a jigsaw puzzle, each net fragment fits to a choice of possible neighboring fragments (and each neuron can be part of several net fragments), so that a virtually infinite number of nets can be formed from a finite set of neurons and net fragments.

According to this principle, a two-phase process admits first a transient initial volley of activity, while in a second phase, most of this activity is silenced so that only mutually supporting neurons stay active. What is responding to the input of a cortical area are thus not individual neurons but rather organized nets. From Gestalt psychology (Wertheimer et al., 1938; Köhler, 1992), we know the visual system can rapidly discern holistic structures, that is, recognize global patterns as arrays of local features that conform consistently to certain “local Gestalt rules” Köhler (1992): from all the local features present in the input, only those are selected by the visual system that fit consistently into a coherent global pattern (coherent in the sense of being composed throughout of overlapping fragments). Thus, local decisions are conditioned on global patterns, whereas global patterns need for their constitution local features. We offer our nets of lateral connections as the mechanism that mediates between local and global levels: local Gestalt rules are implemented by net fragments, and as long as an input pattern can be covered throughout by fragments, the response it elicits is a coherent net.

The two-phase process described is able to remove noise and filter out irrelevant details. It thus avoids the fallacy of early commitment by conditioning neuron activation on integrability into a globally consistent net. An additional effect is that neurons that aren’t receiving afferent input but have sufficient lateral support might get activated, thus complementing figures or textures and allowing the interpretation of occluded objects.

The connectivity necessary for stable second-phase activity is very specific and needs to be generated by learning. The goal is to find and stabilize those patterns of afferent activation that stand out statistically. This is possible with the help of Hebbian plasticity (Hebb, 1949) that strengthens short-range excitatory connections between neurons often co-activated by the input. In the initial stages of learning, before the establishment of supporting connections, learning has to rely on phase one activity, that is, on the actual statistics of input activation. In the course of learning, however, plasticity has to be biased more and more in favor of those neurons that stand out by lateral support, neurons whose activity survives into phase two.

How do our cortical areas compare to the layers of ANNs? Whereas the latter are mainly there to accommodate a hierarchy of feature detectors of higher and higher complexity, already a single cortical area provides for such hierarchy

in the form of nested net fragments of growing size (von der Malsburg et al., 2022b), a hierarchy that extends up to input-activated nets representing entire objects. One role of higher cortical areas is to permit the construction of representations invariant to transformations affecting lower areas (translation, rotation, scaling, etc.), while the connection between variant and invariant representations is established by structured dynamic sets of connections (see below).

Areas are dominated by short-range lateral connections, which abound in the cortex Stettler et al. (2002); Suzuki et al. (2023) but are ignored in multi-layer perceptrons (MLPs) (Prince, 2023) and transformer architectures (Vaswani et al., 2017). Lateral, recurrent connections play important roles both for function and for learning. As for function, they are the basis for the establishment of nets as high-level (e.g., object-global) representations by a collective computational mechanism Forrest (1990), conditioning global and local decisions on each other, as noted above. Such mediation between levels is not possible in ANNs with their purely feed-forward flow of information (which is unavoidable due to their learning principle of backpropagation of error (Rosenblatt, 1962; Linnainmaa, 1970; Rumelhart and McClelland, 1987)). Regarding learning, the restriction to connectivity patterns that are self-consistent (connectivity structure supporting activity states that stabilize this same connectivity structure) reduces the search space of learning by many orders of magnitude and has the potential to produce connectivity patterns representing novel structures never seen as examples. Moreover, the stabilization of connection weights based on local rules (rather than backpropagation through many layers) decisively relaxes precision requirements put on them.

In this work, we present the first implementation of a cortical area structured by fragments and apply it to a simple vision task, showcasing its ability to be robust against noise and perform figure completion. We believe that this conceptual work lays the foundation for a novel learning system that can fulfill complex tasks such as invariant object recognition when scaled up to multiple areas in subsequent research (see Section 2.3). Specifically, we make the following contributions:

- We demonstrate the viability of net fragments as neural code, highlighting their role in enhancing the robustness of computational frameworks.
- We provide a summary of the relevant neuroscientific hypotheses of net fragments and dynamic linking.
- We translate these hypotheses into a computational framework compatible with recent machine learning advances.
- We discuss the introduced concepts and describe how extending it with multiple areas can foster invariant object detection.

2 Related Work

As outlined above, deep neural networks exhibit a deficiency in robustness. In the scientific literature, robustness is typically defined as immunity to noise (additive noise, or subtractive noise in the form of missing information; see Cao et al. (2021); Neururer et al. (2024)) and adversarial attacks (Drenkow et al., 2022), as well as the capability of dealing with ambiguous inputs (Simmler et al., 2021) and generalization within/across domains (e.g., over image translation, rotation, changing lighting, etc.).

This work focuses on analyzing net fragments that are formed in each DNA-based area concerning their robustness to small perturbations. Related work pertaining to this aspect is summarized in Section 2.1, followed by a review of representation learning. The subsequent Section 2.3 examines systems utilizing correspondence-based mapping, which can cope well with ambiguous inputs and exhibit good generalization capabilities. These systems closely align conceptually with the DNA proposed in this work, making them highly relevant for future research, especially for scaling to a multi-area DNA.

2.1 Robustness to Noise

Adversarial attacks exploit deep networks' vulnerability to noise by calculating slight perturbations capable of altering the model's predictions for specific samples (Szegedy et al., 2014; Goodfellow et al., 2015; Papernot et al., 2016a; Carlini and Wagner; Deriu et al., 2022) or perturbations applicable across the entire dataset (Moosavi-Dezfooli et al., 2017). Adversarial attacks can also be devised without direct access to the model and its parameters, for example, by analyzing a model's decision boundary (Brendel et al., 2018; Engstrom et al., 2019) or training an auxiliary model (Chen et al., 2017; Cisse et al., 2017; Sarkar et al., 2017).

Current approaches aim to counter these attacks by modifying the network's architecture to increase robustness (Papernot et al., 2016b; Gao et al., 2017; Ros and Doshi-Velez, 2018), including adversarial examples to the training

data (Moosavi-Dezfooli et al., 2016; Goodfellow et al., 2015), transform the inputs before being processed (Dziugaite et al., 2016; Bhagoji et al., 2018), or adding external models (Meng and Chen, 2017; Xu et al., 2018) or attack detectors (Amirian et al., 2018).

The lack of robustness also manifests itself in poor generalization between domains, where a change in the statistical distribution of data prevents the applicability of a model trained on a source domain to a slightly different target domain (Tuggener et al., 2022; Yan et al., 2024). Addressing such domain shifts typically involves domain alignment (Erfani et al., 2016; Ghifary et al., 2017; Rahman et al., 2020; Sager et al., 2022), meta-learning (Dou et al., 2019; Zhou et al., 2020; Du et al., 2020), or data augmentation (Carlucci et al., 2019; Zhang et al., 2020; Tuggener et al., 2024).

Despite notable advances in improving robustness and preventing adversarial attacks, existing methods primarily address symptoms by reducing the number of parameters (Papernot et al., 2016b), removing specific features (Gao et al., 2017), regularizing gradients (Ros and Doshi-Velez, 2018), augment or extend the training data (Moosavi-Dezfooli et al., 2016; Dziugaite et al., 2016; Tuggener et al., 2024), align domains (Erfani et al., 2016), or meta-learn features (Dou et al., 2019). In contrast, we here attempt to fundamentally rethink the principles underlying learning, network architecture, and information flow, potentially leading to more robust representations. In our experiments (see Section 6), we substantiate the efficacy of the proposed net fragments in effectively mitigating random Gaussian noise and dealing with occluded objects (a type of subtractive noise).

2.2 Representation Learning

Other research relevant to this work employs other design principles to enhance robustness. A crucial principle involves sparse neural activation, which enhances robustness relative to dense representations, as shown for deep neural networks (Guo et al., 2018; Liao et al., 2022; Timpl et al., 2022; Prince, 2023). This is particularly true for binary sparse distributions (Ahmad and Scheinkman, 2019). Element-wise comparison of two vectors, each with only a few activated neurons, proves to be highly robust. Given sufficiently large vector dimensionality, accidental similarity across different binary sparse representations is highly improbable (Ahmad and Hawkins, 2015).

Also in this work, we exploit binary sparse representations to bolster model robustness. In our system, sparsity is not imposed by specific constraints but by design along established principles, including lateral connections (Kothari and Agyepong, 1996; Corchado et al., 2003) and self-organization through local Hebbian updates (Hebb, 1949), coupled with inhibitory mechanisms for activity regulation (Abbott and Nelson, 2000; Luz and Shamir, 2012).

2.3 Correspondence-Based Mapping

A highly robust vision framework for object detection is based on correspondence mapping between pixel patterns within an image and a corresponding prototype (Wolfrum et al., 2008). Here, “corresponding” refers to neurons relating to the same point on the object’s surface. A pivotal concept in this framework are maps implemented as sets of projection fibers (connections that dynamically turn on and off) known as shifter circuits (Anderson and van Essen, 1987; Olshausen et al., 1993) that are composed of maplets (Zhu and von der Malsburg, 2004). These maps are activated when point-to-point correlations are detected between coherent nets in two cortical areas. The composite nets in the two cortical areas explicitly represent spatial relations, and a map between them can only be formed if they agree on this spatial structure. Once a forward mapping to a coherent model of a visual pattern is established, backward mapping to the primary visual domain can stabilize the modeled structure there by complementing missing elements or by lacking to support visual noise elements.

So far, correspondence-based mapping has been applied only to human face detection and identification (Lades et al., 1993; Wolfrum et al., 2008; Fernandes and von der Malsburg, 2015), in which image and object model match with little deformation. For general classes of visual structure, mapping has to tolerate deformation and must be based on the matching of robust feature constellations (Biederman and Kalocsai, 1997). Net fragments can be seen as an attempt to model these (von der Malsburg, 2014, 2018; von der Malsburg et al., 2022a). So far, however, net fragments have only been proposed as a theoretical concept and have not been modeled in any concrete fashion.

In this work our focus is, however, not on correspondence-based mapping, but on the learning of fragment-based net formation, hoping to thereby contribute to the establishment of this neural code as a sound conceptual framework translating concepts of biological vision to applications in next-generation machine learning systems.

3 Net Fragments for Visual Perception

In the following, we restrict the discussion of DNA to the visual system. Light fields captured by the human eyes are translated to retinal images that appear as two-dimensional arrays of activated neurons in the primary visual cortex

(Grill-Spector and Malach, 2004). There, neurons are sensitive to local texture elements within their receptive fields (Grill-Spector and Malach, 2004) and interact through short-range lateral connections (Gilbert et al., 1990). Visual patterns that appear with significant frequency within an image patch lead, through Hebbian plasticity, to the formation of net fragments, as visualized in Figure 1.

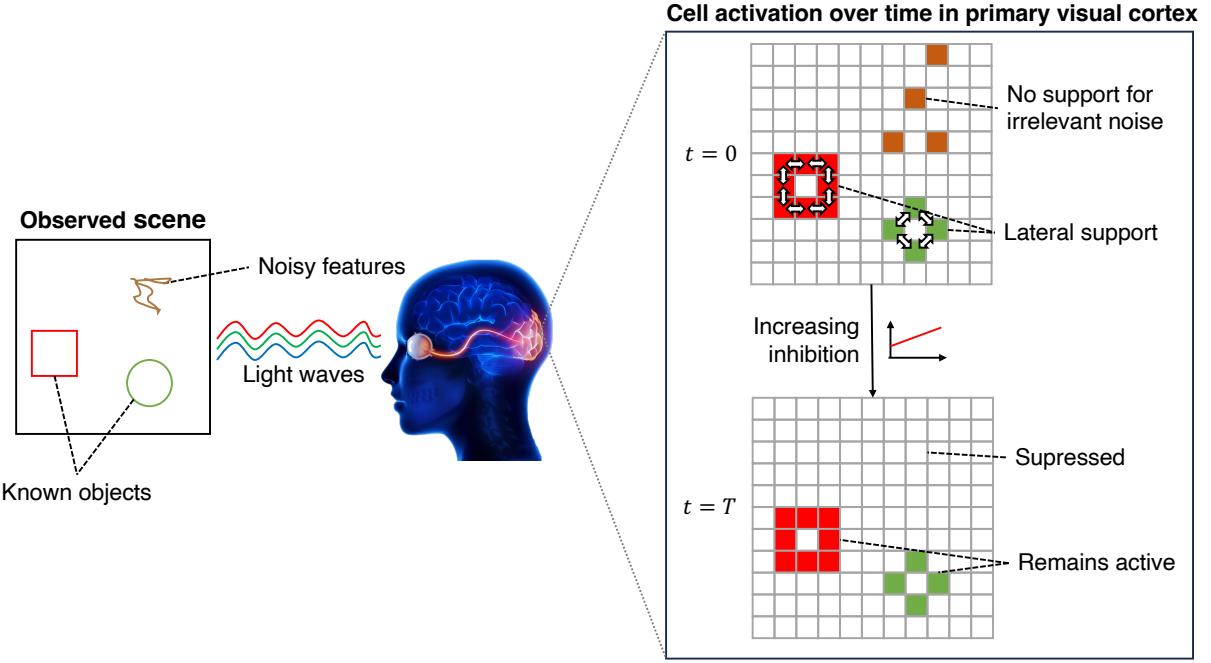


Figure 1: Net fragments in a nutshell: The eyes observe a visual scene and activate neurons in the primary visual cortex. Neurons that support each other inside net fragments remain active, while neurons initially activated by afferent excitation but lacking such support are silenced by inhibition.

Among afferent input-activated neurons, only a few are part of net fragments so as to support each other’s activity by exchanging spikes along excitatory lateral connections (Stettler et al., 2002). Within milliseconds, neurons that have sufficient lateral support are singled out by neural inhibition (Vogels et al., 2011) from all others that also receive afferent input but lack lateral support.

All statistically significant visual textures within image patches with the radius of the range of lateral connections are represented by local connectivity patterns (such as in image compression algorithms (Wallace, 1991) by codebook vectors). To account for the homogeneity of image structure, it can be assumed that the connectivity has translational symmetry (“convolutional” structure) supporting equivariance, that is, the property that image shift gets reflected in activity shift.

A given feature can be part of a large number of patterns. To avoid interference between these patterns, each feature (defined in the primary visual area by a particular receptive field shape in a particular position) is represented by a number of “alternative neurons,” which are free to carry different connections¹. In this way, features can vary their connections to other features by activating different alternative neurons: they are connected by “dynamic links.” For more details on alternative neurons, please refer to Appendix A.

To project net fragments and coherent nets out from the mass of all neurons that are individually activated by the sensory input needs dynamic processing: After neurons are activated by afferent input, they are subjected to progressively increasing inhibition, which eventually silences all those of them that don’t have sufficient lateral support. As many fragments can survive inhibition only if supported by overlapping other fragments, inhibition-resistant responses constitute nets that are coherently composed of overlapping fragments.

These fragment-composed nets differ fundamentally from associative memories as formalized, for instance, by Hopfield (1982); Cohen and Grossberg (1983). In these, entire neural patterns, “assemblies,” are stored (in a single step) and

¹ Alternative neurons may be seen as analogous to transformer’s Vaswani et al. (2017) token embedding vector components, which support the different connections between tokens represented by the query-key mechanism.

can be retrieved only as a rigid whole, while nets are flexibly composed of overlapping fragments (which have been learned in a statistical process) and are flexibly assembled to model the input. We discuss this distinction in more detail in Appendix A.

4 Computational Framework

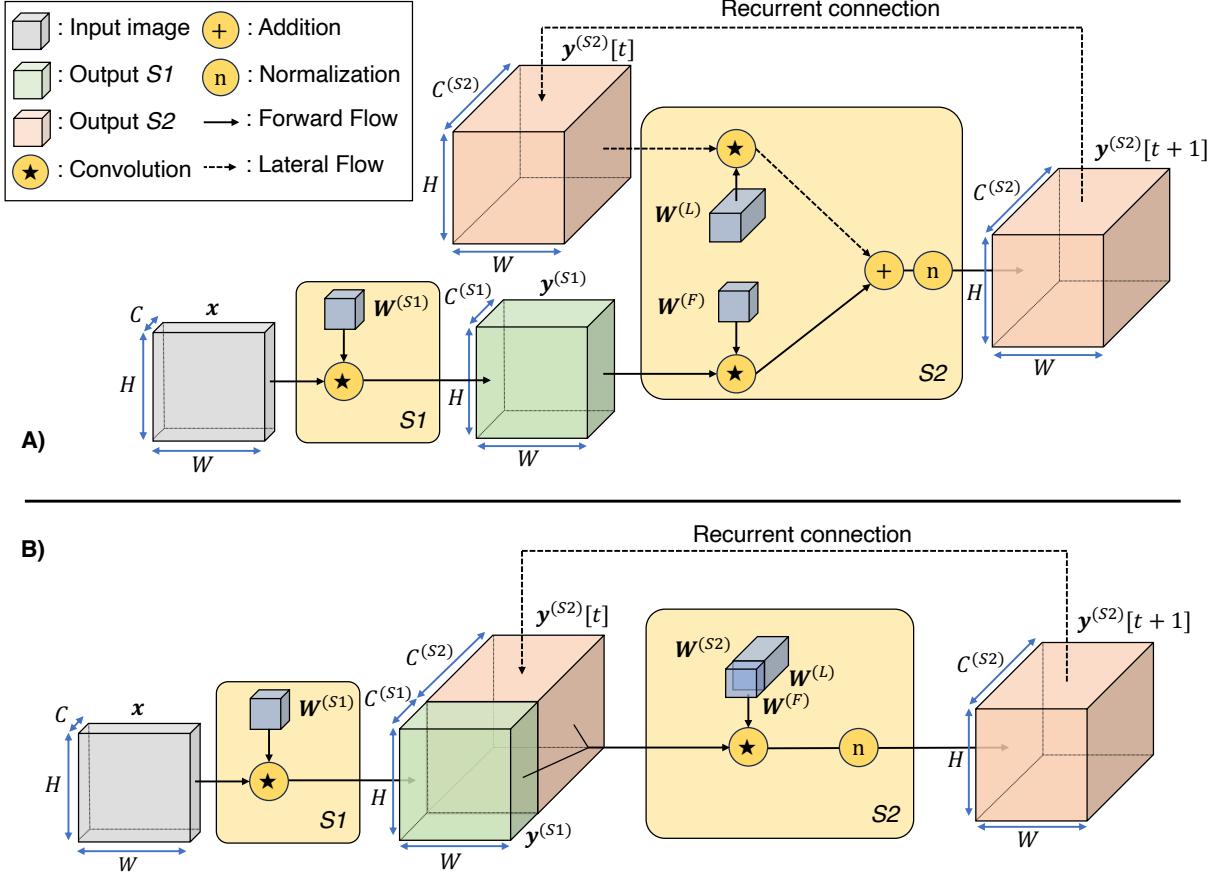


Figure 2: The dynamics of a single area within the proposed DNA: The input image is fed into stage $S1$ to obtain feature activations. The feature activations are then processed by stage $S2$, together with activations from neurons within the same layer. In the upper part A, two distinct matrices are used to model the forward connections from $S1$ to $S2$ and the lateral connections within $S2$. In the lower part B, the matrices and input signals are concatenated, resulting in one bigger weight matrix.

We here restrict ourselves to modeling this functionality within a single cortical area, although the significance of coherent nets in primary sensory areas can be fully appreciated only in the context of different interacting areas.

A single area of a DNA model comprising net fragments is shown in Figure 2(A). Our system is based on an input layer representing observed images by simple feature detectors (which are perhaps to be localized within layer IV of the primary visual cortex, see, for instance, Usrey et al. (2003)) and two layers, *Stage 1* (denoted as $S1$) and *Stage 2* (denoted as $S2$) whose neurons are thought to be also localized in the same primary visual cortex area, presumably within layers II and III.

Feature Extraction. The system's input is a retinal 2D image represented by variables $\mathbf{x}_{k,i}$ (gray box in Figure 2), where $k \in (1, \dots, C)$ denotes the channel index of the image ($C = 1$ for grayscale, $C = 3$ for RGB), and the composite index $i = (h, w) \in (H \times W)$ stands for vertical and horizontal position of pixels within the input image.

Such an image is used as afferent input to the feature extraction neurons in $S1$ and first increases their “membrane potential” $\mathbf{a}^{(S1)} \in \mathbb{R}^{C^{(S1)} \times H \times W}$ and potentially triggers their binary activation state $\mathbf{y}^{(S1)} \in \{0, 1\}^{C^{(S1)} \times H \times W}$ (green

box in Figure 2). The membrane potential $\mathbf{a}^{(S1)}$ is calculated by applying a convolutional operation (LeCun et al., 1989) (yellow box $S1$ in Figure 2):

$$\mathbf{a}_{c,j}^{(S1)} = \sum_{k \in C} \sum_{i \in l_j} \mathbf{W}_{c,k,j-i}^{(S1)} \cdot \mathbf{x}_{k,i} \quad (1)$$

Here, $c \in C^{(S1)}$ denotes the feature channel, the compound index j , expanded as $j = \{h, w\}$, designates again spatial position $h \in H, w \in W$, and the compound index $i = \{h, w\}$ is running through all (h, w) -pairs in the two-dimensional range $l_j = (h, \dots, h + h^{(W1)}; w, \dots, w + w^{(W1)})$ where $h^{(W1)}, w^{(W1)}$ denote the size of the convolutional filter. The size of the convolutional kernel used for feature extraction is $\mathbf{W}^{(S1)} \in \mathbb{R}^{C^{(S1)} \times C \times h^{(W1)} \times w^{(W1)}}$.

Feature Binarization. The membrane potential $\mathbf{a}^{(S1)}$, together with a bias parameter $b^{(S1)}$, determines the binary output of the neurons in $S1$, denoted as variable $\mathbf{y}_{c,j}^{(S1)} \in \{0, 1\}^{C^{(S1)} \times H \times W}$, where 1 stands for firing, 0 for silence. The binary neuron model with the input-output function $B : \mathbf{a}, b \rightarrow \mathbf{y}$ is defined as

$$\mathbf{y}_{c,j}^{(S1)} = B(\mathbf{a}_{c,j}^{(S1)} - b^{(S1)}) = \begin{cases} 1, & \text{if } \mathbf{a}_{c,j}^{(S1)} - b^{(S1)} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

with the same bias parameter $b^{(S1)}$ for all neurons acting as a firing threshold (cells fire if the membrane potential is above this threshold).

Dynamics. Stage $S2$ builds net fragments by allowing active neurons to support laterally connected neurons over T time steps (depicted as recurrent connection in Figure 2). A sample image is introduced into the system at time step $t = 0$ to activate neurons and remains unchanged until time step $t = T$. Throughout this period, neuronal activity undergoes sparsification due to growing inhibition. Through the sparsification process, the system forms an object representation by leaving active a set of neurons each of which is supported by active neighbors, that is, a net.

Short-Range Lateral Connections. Although typical convolutional neural networks (Fukushima, 1980; Waibel et al., 1987; LeCun et al., 1989) apply convolution operations to establish *feed-forward* connections between different layers, we employ them here to implement short-range *lateral* connections within the same layer, their symmetry permitting a pattern to manifest itself at any spatial location based on corresponding net fragments. Given lateral connections, the neural activity of the previous time step $t - 1$ is accessed to calculate a neuron's activity at time t . In contrast to recurrent networks like LSTMs (Hochreiter and Schmidhuber, 1997), where a new token is input at each time step, we iterate over the same input while updating the internal state.

The membrane potentials $\mathbf{a}_{c,j}^{(S2)}$ of the second stage neurons get excited by the (constant) signals of the previous stage $S1$ and iteratively by lateral excitation from within $S2$. Before digitization, they are modified by saturation, normalization, and inhibition (yellow box $S2$ in Figure 2). Before those modifications, its raw form is computed as:

$$\mathbf{a}'_{c,j}^{(S2)}[t] = \underbrace{\left(\sum_{k \in C^{(S1)}} \sum_{i \in l_j} \mathbf{W}_{c,k,j-i}^{(F)} \cdot \mathbf{y}_{k,i}^{(S1)} \right)}_{\text{Forward Connection}} + \overbrace{\left(\sum_{k \in C^{(S2)}} \sum_{i \in l_j} \mathbf{W}_{c,k,j-i}^{(L)} \cdot \mathbf{y}_{k,i}^{(S2)}[t-1] \right)}^{\text{Lateral Connection}}. \quad (3)$$

Similar to $S1$, $c \in C^{(S2)}$ denotes the feature channel of $S2$, the compound index j expands to $j = \{w, h\}, w \in W, h \in H$, and the compound index i is running in the range $l_j = (h, \dots, h + h^{(W2)}; w, \dots, w + w^{(W2)})$, with $h^{(W2)}$ and $w^{(W2)}$ being the size of the convolution kernel.

The forward connections are denoted as $\mathbf{W}^{(F)} \in \mathbb{R}^{C^{(S2)} \times C^{(S1)} \times h^{(W2)} \times w^{(W2)}}$ and the lateral connections as $\mathbf{W}^{(L)} \in \mathbb{R}^{C^{(S2)} \times C^{(S2)} \times h^{(W2)} \times w^{(W2)}}$, where $C^{(S2)}$ is the number of channels in $S2$, including the set of all feature channels and its alternative channels, i.e. $C^{(S2)} = n_a \cdot C^{(S1)}$ where n_a denotes the number of alternative channels per feature type (the alternative channels handle the alternative neurons that are required to prevent cross-talk between fragments, see Appendix A). In the case of the forward connections, the kernel size determines which of the neurons $\mathbf{y}^{(S1)}$ in stage 1 can be connected to a neuron $\mathbf{y}^{(S2)}$ in stage 2 (red box in Figure 2), while the kernel size of the lateral connections

determines which neurons can be connected within stage 2. The kernel $\mathbf{W}^{(F)}$ is initialized such that the sets of equivalent neurons in $S2$ copy the activity (and with that the feature type) of the corresponding neuron in $S1$, and $\mathbf{W}^{(L)}$ is initialized with zeros except for self-coupling of neurons, with value 1 (details in Appendix B). The weights are updated using Hebbian Learning (details in Appendix C).

For the first time step $t = 0$, when $\mathbf{y}_i^{(S2)}$ is undefined, we initialize the $S2$ neurons with zeros, i.e., the neurons are silent and do not provide lateral support:

$$\mathbf{y}_i^{(S2)}[t = 0] = 0 \quad (4)$$

Single Weight-Kernel Implementation. This entire process is visually depicted in the upper part (A) of Figure 2. For simplicity in our algorithm, and since we use the same kernel size for the forward and lateral connections, we stack the two kernels $\mathbf{W}^{(F)}$ and $\mathbf{W}^{(L)}$ as $\mathbf{W}^{(S2)} = (\mathbf{W}^{(F)}; \mathbf{W}^{(L)})$ as well as the inputs $\mathbf{y}_{k,i}^{(S1)}$ and $\mathbf{y}_{k,i}^{(S2)}[t - 1]$ as shown in Figure 2(B), leading to the equivalent formulation:

$$\mathbf{a}'_{c,j}^{(S2)}[t] = \sum_{k \in C^{(S1)}} \sum_{i \in l_j} \mathbf{W}_{c,k,j-i}^{(S2)} \cdot (\mathbf{y}_{k,i}^{(S1)}; \mathbf{y}_{k,i}^{(S2)}[t - 1]) \quad (5)$$

The net fragments are implicitly contained in this processing as the active neurons whose activity state is stored in $\mathbf{y}^{(S1)}$ and their connection to other neurons.

Inhibition. The raw neural excitation \mathbf{a}' computed in eq. 3 (or eq. 5) is to be subjected to saturation, inhibition, and normalization so it can be binarized using a pre-defined threshold. These steps of processing the raw membrane potential \mathbf{a}' of neurons proved necessary in our simulation experiments to achieve the qualitative function we aimed for. The details to obtain the normalized activations \mathbf{a}^{norm} in the range $(0, 1)$ are described in Appendix D. The differences between activation levels of neurons are accentuated by setting

$$\mathbf{a}_{c,j}^{(S2)}[t] = \max \left(\left(\mathbf{a}_{c,j}^{\text{norm}}[t] \right)^\gamma, 0 \right), \quad (6)$$

The coefficient γ regulates the inhibition strength and increases with the time-steps of the inner loop according to $\gamma = 1.2 + 0.2t$, making it harder for neurons to fire (as $\mathbf{a}_{c,j}^{\text{norm}}[t] \leq 1$). The membrane potential is higher for neurons that are part of coherent fragments (valid patterns) as they receive lateral support, while invalid activations are less supported. While the inhibition coefficient increases over time, many neurons eventually get suppressed, leaving only those active that support each other and form a consistent net. Finally, the membrane potential is converted to a binary activation, neurons becoming active when their inhibited membrane potential is above the activation bias $b^{(S2)}$, which serves to control the firing rate of neurons:

$$\mathbf{y}_{c,j}^{(S2)}[t] = \text{B} \left(\mathbf{a}_{c,j}^{(S2)}[t] - b^{(S2)} \right) \quad (7)$$

The final output $\mathbf{y}^{(S2)}[t = T]$ is the lateral connection-supported neuronal activation in which, in contrast to the initial activation, unsupported activations are suppressed. To allow features to be part of multiple net fragments and represent various patterns, they are represented by a set of alternative neurons (details in Appendix A). All neurons within such sets are free to learn different lateral connections to other neurons. They are connected by distinct channels within the convolutional matrix and engage in competition, with only one alternative channel gaining activation at each spatial location while the remaining neurons are suppressed (set to zero). This selection process is described in the Appendix E.

5 Experiments

We evaluate our framework by assessing the efficacy of net fragments to filter out random Gaussian noise, the capacity to reconstruct occluded objects, i.e., the efficacy of dealing with subtractive noise, and its generalization capacity by composing small net fragments into object structures not observed during training.

5.1 Dataset

The dataset comprises binary images measuring (32×32) pixels, each sample depicting a straight line going through the image center, starting and ending 2 pixels from the image boundary (leading to 59 distinct lines of different angles).

The images are generated when required and their set may vary from one epoch to another. During each training cycle, we randomly generate 300 image instances (each line about 5 times). During evaluation, we sample each of the 59 distinct lines exactly once. In contrast to training, the samples used for evaluation comprise local distortions in the form of additive Gaussian noise and missing line segments (subtractive noise), and we evaluate the net fragments' capability to remove these.

We introduce additive Gaussian noise to the afferent input in stage $S2$, thereby simulating an image structure that has already survived the filtering of feature kernels in $S1$. To that end, we probabilistically flip neurons within each channel with a probability of up to 20%. Since we use four feature channels in our experiments, this corresponds to a probability of $1 - (1 - 0.2)^4 = 59\%$ that a neuron is flipped at any spatial location. To evaluate subtractive noise, we create discontinuous lines by deleting a line segment in the middle.

To demonstrate the generalization capability of DNA areas, we generate validation samples that differ from the ones used during training. Specifically, we produce kinked lines by generating four random points within the image and draw straight lines between these points.

6 Results

In the following, we show that the proposed net fragments can suppress Gaussian noise in Section 6.1 and can reconstruct removed pixels, see Section 6.2. A description of the feature extraction mechanism to obtain $y^{(S1)}$ is in Appendix F, and details about the used parameters are in Appendix G. Additionally, Appendix H contains visualizations of the average support strength received by cells, and Appendix I a quantitative evaluation of suppressed noise.

6.1 Filtering Gaussian Noise

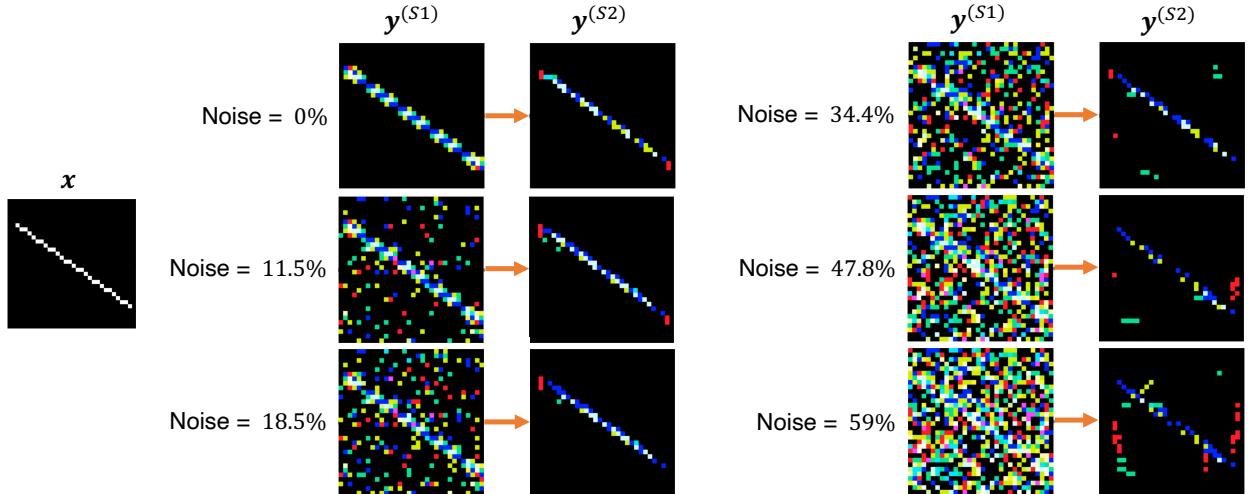


Figure 3: Noise-corrupted feature activations $y^{(S1)}$ alongside the corresponding output $y^{(S2)}$, in which noise is reduced due to suppression of neurons not supported by lateral connections of net fragments. In each image, the background is shown in black, and the four different features in distinct colors. The plots are created using a bias $b^{(S2)} = 0.5$ and an inhibition coefficient $\gamma = 1.2 + 0.2t$.

The illustration in Figure 3 visually demonstrates the system's effective capability of filtering out isolated noise or small clusters thereof. Noise of varying degrees (from 0% to 59% noise probability at each spatial location - see Appendix 5.1) is introduced in an area's afferent input, from which a significant portion is removed when constructing net fragments. While introducing up to 47.8% noise, remarkably consistent outputs $y^{(S2)}$ are generated. As additional noise is introduced, more undesired neurons receive support and persist in their activity. However, the overarching pattern remains distinguishable even when subjected to noise levels of up to 59%. In Appendix I, we quantitatively confirm the efficacy of noise filtering by measuring precision, recall, and noise filtration rate for various parameter settings, showing that lines remain well distinguishable for up to 59% added noise.

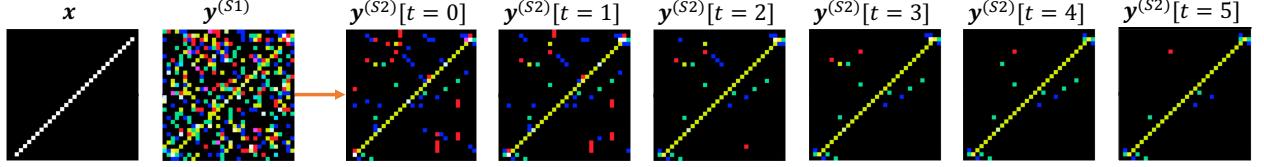


Figure 4: The filtering of (simulated) noise over time: Already at time step $t = 0$, most of the noise is suppressed. However, when inhibition increases over time, even more noise is filtered out in the course of iteration. The plot is created after adding 10% noise to each feature channel (sing a bias $b^{(S2)} = 0.7$ and an inhibition coefficient $\gamma = 1.2 + 0.2t$).

Figure 3 visualizes that neural activity simulating visual clutter without coherent structure is removed after processing for 5 time-steps. Figure 4 depicts this systematic filtering over time. The additionally introduced clutter doesn't activate overlapping net fragments and cannot withstand increasing inhibition. The figure illustrates a substantial noise reduction already at time step $t = 0$ due to the filtering by matrix $\mathbf{W}^{(F)}$ and initial inhibition. In each subsequent time step, additional clutter is removed. This reduction is due to increased inhibition and falling support as a consequence of the drop-out of supporting neurons. We thus observe a chain reaction of gradual suppression of incoherent structure as well as the preservation of net structure on the basis of internal support.

6.2 Reconstructing Subtractive Noise

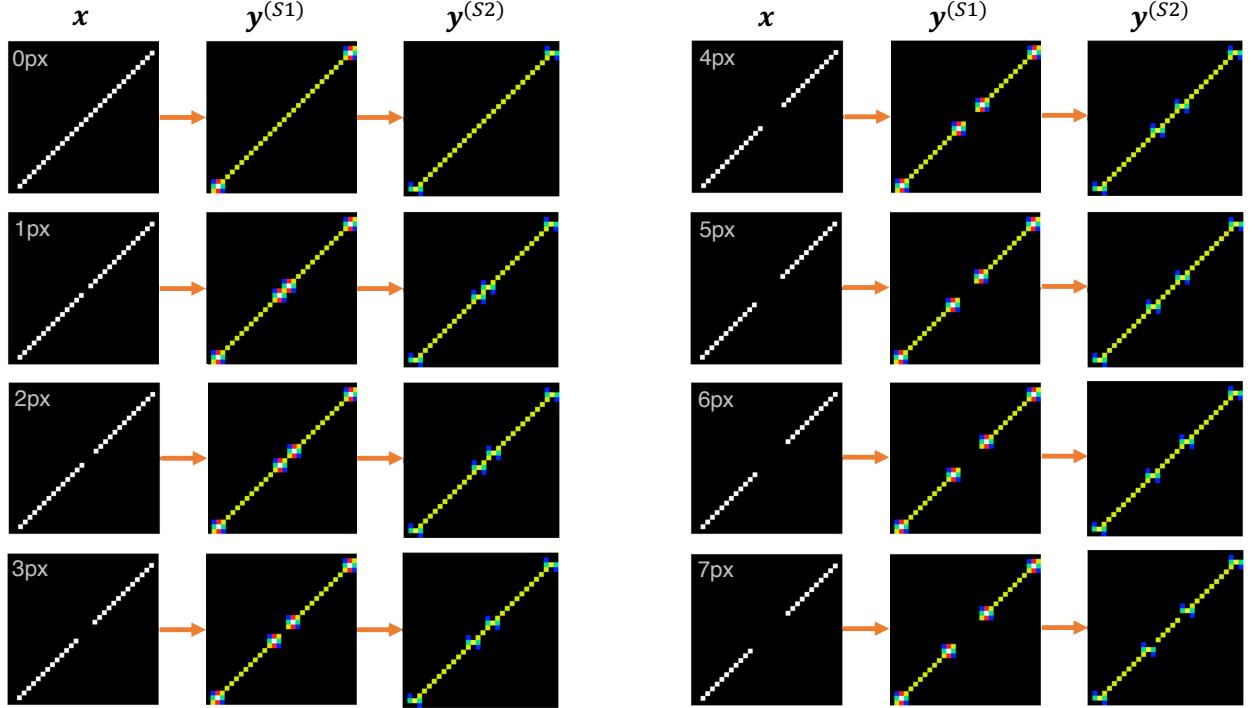


Figure 5: Partial occlusion as subtractive noise: Input images of diagonal lines with different numbers of missing pixels, denoted in the top left corner of the panels (left column), corresponding feature activations (middle column), and final output neuron activations. Activation bias $b^{(S2)} = 0.5$, inhibition coefficient $\gamma = 0.6 + 0.2t$.

If a substantial number of neurons activate and support other neurons within a net fragment, inactive neurons can receive significant support, leading them to switch on and encouraging figure reconstruction. Figure 5 shows how inactive neurons activate and thus demonstrates that net fragments can deal with subtractive noise (occluded patterns). This reconstruction works only reliably for up to 3 missing pixels (see Appendix I). In some cases, as in this Figure, the line is fully reconstructed for up to 6 removed pixels and partly reconstructed for 7 pixels.

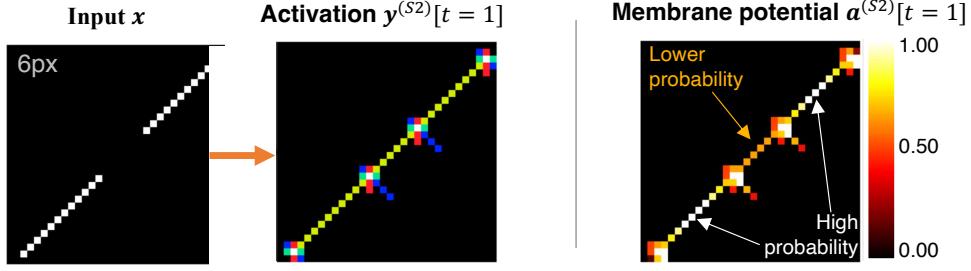


Figure 6: Diagonal line with 6 missing pixels in the center (shown on the left), the corresponding output activity (in the middle), and the membrane potential $a^{(S2)}$ displaying the “activation probabilities” (on the right) after the first time step. The activation probability is close to 1 where a line has been observed and around 50% in the middle where no line was observed but expected. Activation bias $b^{(S2)} = 0.5$, inhibition coefficient $\gamma = 0.6 + 0.2t$.

Membrane Potential Represents Expected Feature Activation. When dealing with missing pattern elements, the membrane potential map $a^{(S2)}$ plays a crucial role. The potential is higher at spatial locations where line features are observed and lower where they are not observed but expected. Consequently, the network can, to some extent, handle this ambiguity and model it based on the activation map, as shown in Figure 6. Modeling this ambiguity enables having different internal interpretations, such as whether “it is an interrupted line that should be reconstructed” or “there are two co-linear line segments.” We find that the activation map is especially reliable during the first time step, where it is calculated based on the observed input and the network’s expectation about local patterns. Since local interactions can trigger or deactivate neurons, the activation map changes over time, reducing its interpretability.

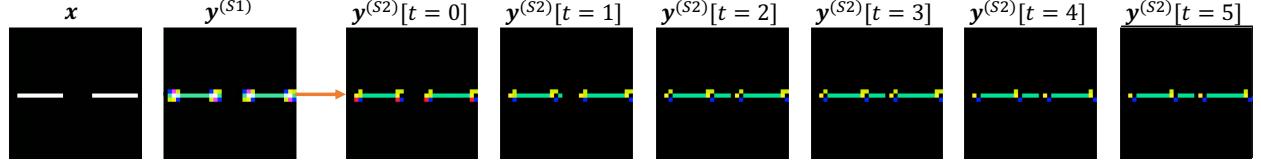


Figure 7: Reconstruction of a horizontal line interrupted by 7 pixels: The feature activation and the initial output activity at $t = 0$ are interrupted but reconstructed within the two subsequent time steps ($t = 1, 2$). Afterward, some artifacts triggered at line ends are reduced due to increasing inhibition. Activation bias $b^{(S2)} = 0.5$, inhibition coefficient $\gamma = 1.8 + 0.1t$.

Increasing Inhibition Over Time Stabilizes Patterns. The reconstruction process unfolds across multiple sequential steps, as illustrated in Figure 7. Initially, at time step $t = 0$, the net fragments represent the interrupted line. Subsequently, during the second time step ($t = 1$), some of the removed pixels are reconstructed, and shortly after, at $t = 2$, the entire line is represented by the net fragments. Thus, reconstruction can persist even amid increasing inhibition, given that inactive neurons receive adequate support. This example underscores the necessity for multiple time steps, essential not only for noise removal but also for the reconstruction of features, as neurons can undergo flipping in each time step, fostering a dynamic interplay where the flipping of certain neurons encourages others to follow.

6.3 Compositionality

A single DNA area models a feature hierarchy through nested net fragments of increasing size (von der Malsburg et al., 2022b). In this study, we demonstrate that these higher-level nets can be composed of local features (small net fragments), even though the underlying patterns had never been observed during training. We show this on the example of curved lines, which share only local features with the straight lines of the training set.

Figure 8 depicts four samples of curved lines with the corresponding neuronal activation within an area, once without noise and once with 18.5% noise added to its afferent input. The cortical area is able to represent the curved lines as composite of net fragments learned from straight lines. Added unstructured clutter is efficiently removed, leading to robust input representation.

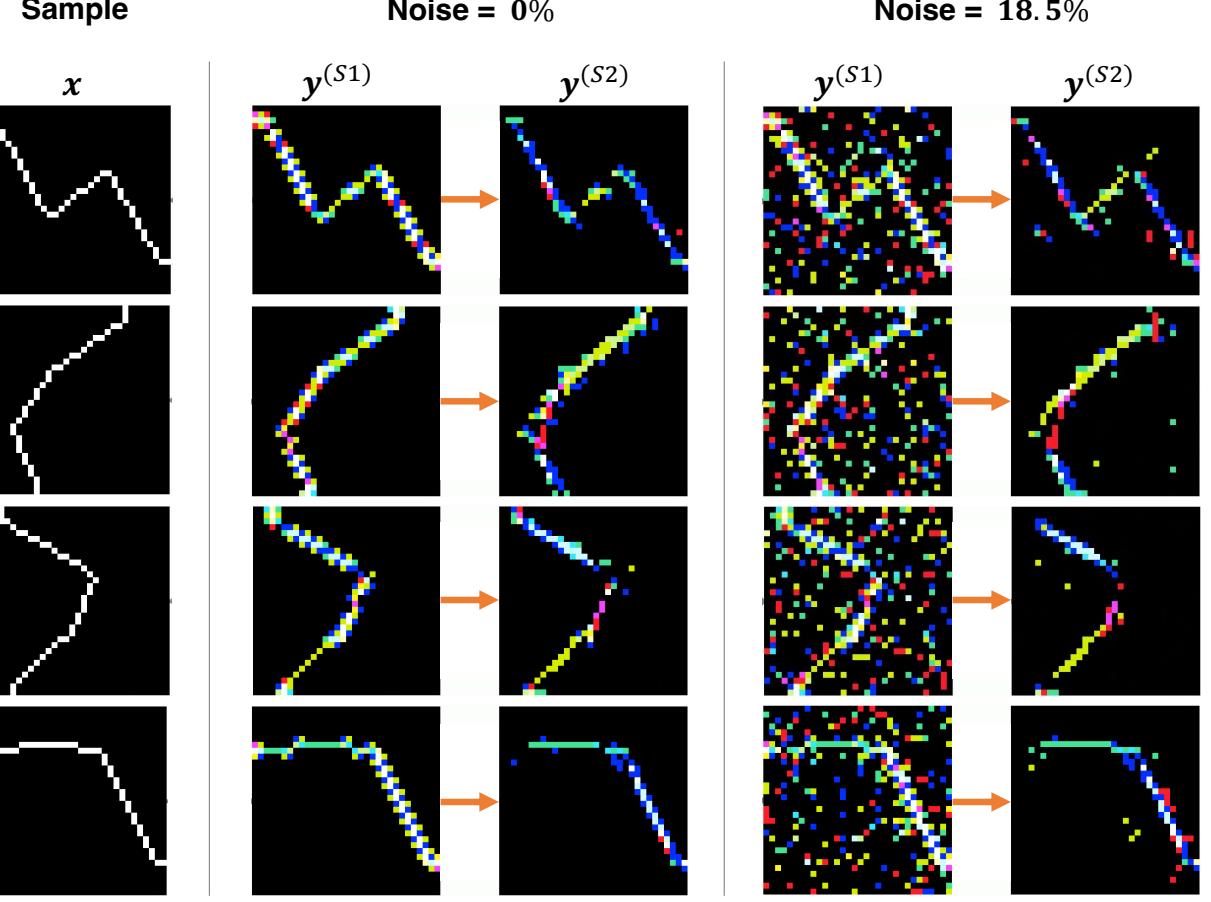


Figure 8: Input images of curved lines are shown on the left, the second and third columns show the corresponding feature and output activation without noise, and the fourth and fifth columns show the same feature and output activation with noise corruption (activation bias $b^{(S2)} = 0.3$, inhibition coefficient $\gamma = 0.6 + 0.1t$).

7 Discussion and Conclusions

The concepts proposed here may be compared to established neural theory. Nets are based on associative learning but differ fundamentally from Hebbian assemblies (see Appendix A) as described in (Hopfield, 1982; Cohen and Grossberg, 1983; Papadimitriou et al., 2020), which are monolithic and without inner structure, whereas nets can represent an infinity of patterns based on shared fragments and can deal with new entities based on homeomorphy between common compositional structure, thus enabling the systematicity and productivity characterizing cognitive operations (Fodor and Pylyshyn, 1988; von der Malsburg, 2023). In comparison to multi-layered perceptrons, coherent nets replace hierarchies of independent feature neurons by net fragments, share with them the potential for equivariance (shift symmetry), but in distinction to them support the construction of an invariant representation that doesn't need to be learned object-by-object (see next section). With the transformer architecture (Vaswani et al., 2017), nets share the provision for finding non-local patterns through on-the-fly computation of lateral links.

A decisive difference between DNA and all deep-learning systems (LeCun et al., 2015; Schmidhuber, 2015; Stadelmann et al., 2019) is the mode of learning. ANNs typically rely on backpropagation of error where the goal is reaching consistency (minimal error) between system output and teaching signals. With DNA models, the goal is mutual consistency at each neuron, between the excitatory connections converging there, this consistency being measured by the probability for each connection to reach a highly active neuron when carrying a signal itself. Of all possible connectivity patterns, only a minute fraction has this property of self-consistency. This condition of self-consistency acts as a selection criterion for eligible networks and thus powerfully reduces the search space of learning.

Net fragments and the structured nets they build form a basis for the representation of hierarchical structures. They are effective for feature filtering and for preventing early commitment, which may be of fundamental importance for improving the robustness of current machine learning systems. So far, a neural code based on net fragments has merely been proposed as a theoretical concept (von der Malsburg et al., 2022b), while this study pioneers the implementation of net fragments in a concrete simulation.

Our experiments demonstrate that the proposed DNA architecture builds net fragments that effectively stabilize patterns, enhance robustness by eliminating visual clutter, and can reconstruct removed parts of known patterns. The DNA model can handle additive noise exceptionally well and eliminates up to 96% of it while preserving the initial pattern, even in scenarios where up to 59% of all locations are affected. Figure completion works well for line interrupts for segments of up to 3 pixels, while for larger interruptions, the reconstruction rate decreases considerably. One reason for this is certainly due to the nature of the used 1D pattern: as it involves straight lines with a width of one pixel, typically, only a fraction of the neurons within the local 2D neighborhood of size $h^{(W^2)} \times w^{(W^2)}$ are active. With a larger line interrupt, the proportion of active neurons compared to neighboring neurons is vanishingly small, which makes reconstruction more difficult. It is expected that reconstruction will be much more robust with two-dimensional patterns.

We posit that this work, which focuses on the internals of a single cortical area, allows for scaling to multiple areas that will allow the implementation of more complex tasks (see next section). Nevertheless, although this work is intended to be a step towards implementing a robust learning framework and the proposed neural code exhibits promising characteristics, it still has weaknesses that should be addressed in future work. One such weakness is dependence on manual adjustment of the parameters b and γ . The mechanism of signal control is biologically implausible. It should be replaced by a more elegant, theoretically justified, and self-adjusting mechanism.

8 Future Work

One important purpose of handing structure from one area to another is generalization, a prime example for which in vision is object recognition invariant to translation (and wider transformation groups). While our system only includes one “cortical” area, another area (that would correspond to the cortical inferotemporal cortex (Ito et al., 1995)) could contain invariant object models. In the context of DNA, we see invariant recognition as realized by a process in which an object-representing net in a primary sensory area A forms and activates a homeomorphic net in a secondary area B . Two nets are homeomorphic if they contain the same feature types in the same spatial arrangement. Invariance is achieved if differently transformed mutually homeomorphic nets in A activate the same net in B under a homeomorphic mapping. Inter-areal homeomorphic maps are based on feature type-preserving connections, which in the brain are realized by axonal fibers². In fact, connections should not only be feature type-preserving but also restricted to pairs of corresponding alternative neurons in A and B , such that they carry not only feature information but also information on connectedness³. The generation of the net in B is again the result of a two-phase process, the afferent connections from A to B transiently activating many neurons in B , most of which getting silenced in phase two, leaving only those neurons active that not only got afferent input but are also supported by lateral excitation within B . The net thus formed in B is invariant to homeomorphic transformations in A . Object classification, the identification of the net in B as a particular object type or recognition of individual objects needs a further area C containing a fixed net that is activated based on approximate homeomorphy with the invariant net in B .

The result of the recognition process is the formation of a larger net composed of a net in A , a net in B , and one-to-one connections between corresponding neurons in A and B . The self-consistency of this larger net is the basis for learning all participating connections, as all connections are stabilized (or generated) by their success in predicting activity in their target neuron.

Through inverse mapping, these stabilizing interactions can be re-imported to the previous area and thus help to discriminate the corresponding nets there from noise and clutter and bridge larger partial occlusions, which would realize the idea of predictive coding Keller and Mrsic-Flogel (2018).

An obvious first step in preparation for a larger-scale system of interacting areas is, however, the training of a single primary-sensory area with natural images to capture as net fragments all textures that occur in them with significant frequency. This project should include the extraction of such different sensory qualities as stereo depth, local motion or surface discontinuities. Net fragments and nets based on these would then have a chance to relate to intrinsic surface properties such as form.

²With the help of several intermediate areas, the number of necessary connections can be brought down to realistic values (Anderson and van Essen, 1987; Olshausen et al., 1993; Wolfrum et al., 2008)).

³To further ease the formation of homeomorphic maps in B , a mechanism for mutual support between mapping-compatible connections has been proposed and successfully applied (see Section 2.3).

Funding

This research has been funded by the Canton of Zurich, Switzerland, through the Digitalization Initiative of the Canton of Zurich (DIZH) Fellowship project “Stability of self-organizing net fragments as inductive bias for next-generation deep learning.”

Data Availability Statement

The code to generate the data as well as to reproduce the results can be found after the publication of this work on Github at <https://github.com/sagerpascal/dynamic-link-architecture>.

Appendix

A Net Fragments vs. Associative Memory, Alternative Neurons

Learning of lateral connections to form net fragments shares aspects with associative memory (Hopfield, 1982; Cohen and Grossberg, 1983). There are, however, important differences. In an associative memory system, neural patterns are stored at once in their entirety with the help of synaptic plasticity, any pair of active neurons within a pattern strengthening their mutual excitatory connections. The resulting connectivity turns each stored pattern into an attractor state that can be recalled (dynamically activated) starting from initial states that come near to it. Such stored and dynamically stabilized neural patterns are often called “assemblies” (Hebb, 1949), and are discussed to this day as fundamental to the function of the brain, see for instance (Papadimitriou et al., 2020).

There are two problems with the concept of associative memory. One is rather fundamental in that a pattern is monolithic and can be recalled only as a rigid whole. This is due to the indiscriminate stickiness with which *any* pair of simultaneously active neurons is tied together by synaptic plasticity in the one-time storage process. Fragments, in distinction, are formed in a protracted statistical process in which connections are gradually established between neurons that are co-active with statistical significance. Due to the spatial deformation of visual patterns, this can only be the case for neurons relating to neighboring visual points. DNA models distinguish insignificant from significant connections, only the latter being permitted to grow, which they do under competition until reaching equilibrium. As a result, the system doesn’t store rigid global patterns but is able to dynamically stabilize an unlimited number of global states that are composed combinatorially out of overlapping fragments.

The other, more technical, problem of associative memory is that stored states have to be statistically independent (“orthogonal”) to avoid cross-talk between them. (In a simple example of cross-talk, a neuron c_1 gets connected at different times with neurons c_2, c_3, c_4 , although the pattern $\{c_1, c_2, c_3, c_4\}$ doesn’t occur with statistical significance during learning. As a result, c_1 may be activated erroneously if $\{c_2, c_3, c_4\}$ is active as part of a larger pattern.) Sparsity (according to which a very small fraction of all neurons are active within any given pattern) has been proposed as a general means to avoid cross-talk and to ensure statistical independence of stored patterns (French, 1999).

DNA models are freed from this constraint by the introduction of “alternative neurons,” neurons that are activated by the same input feature but are free to learn different lateral connections to other neurons. (In our example, neuron c_1 would be replaced by a set $\{c_1, c_{1'}, c_{1''}\}$ of alternative neurons, which could then independently learn the connections $c_1 - c_2, c_{1'} - c_3$ and $c_{1''} - c_4$, thus avoiding the above cross-talk.) In the biological case, there is ample room for alternative neurons. There are, for instance, a hundred times more neurons in the primary visual cortex compared to the number of fibers coming out of the retina (Leuba and Kraftsik, 1994).

B Weight Initialization

The forward connections $\mathbf{W}^{(F)}$ are initialized so that the activations of the $S1$ neurons are copied into the $S2$ neurons in the same position. This is done by setting the connections at the center of the kernel (at position $(h^{(W2)}/2, w^{(W2)}/2)$) to 1 if $c_{in} = \lceil c_{out}/n_a \rceil$, where c_{in} denotes the index of the input channel, c_{out} the index of the output channel, and n_a the number of alternative cells.

The lateral connections $\mathbf{W}^{(L)}$ are initialized with zeros except for the self-coupling of neurons, which is set to 1. When Hebbian plasticity is employed on the weight matrix $\mathbf{W}^{(L)}$, these connections ought to grow (as neurons tend to keep their activation state between time steps), but due to the constraint of keeping all connection weights within the range (0, 1) (refer to equation 8), they stay put at 1.

C Hebbian Plasticity

After processing for T time steps both matrices $\mathbf{W}^{(F)}$ and $\mathbf{W}^{(L)}$ undergo Hebbian updates (Hebb, 1949):

$$\mathbf{W}^{(S2)} := \min \left(\max \left(\mathbf{W}^{(S2)} + \alpha \cdot \rho_{avg}, 0 \right), 1 \right) \quad (8)$$

where α represents the learning rate, and ρ_{avg} denotes the average correlation between input and output. This average is determined for a given activation over all instances of a given pair of neurons c, c' in different positions of the image domain. To compute ρ_{avg} , we count the number of instances in which both neurons are on, subtract from it the number of instances where one neuron is on, and the other is off, and divide that sum by the number of all positive and negative instances. The actual modification eq. (8) is applied once after the last iteration over T timesteps to increase stability against fluctuation of the activities $\mathbf{y}^{(S2)}[t]$ in the course of iterations.

The network employs shared connections, formulated as convolutional kernels, between input patches and output neurons, making the building of fragments translation equivariant. Consequently, the same lateral connection is applied across multiple output and input neurons. The decision to update this connection is based on the average correlation ρ_{avg} between all input and output neurons it connects. If the resulting average is positive, the corresponding connection increases by α as it connects more simultaneously active neurons than neurons that fire disjointly. Conversely, if the average correlation is negative, indicating more disjoint firing, the connection strength is reduced by α .

D Normalization

We use two normalization steps to confine the neuronal activations $\mathbf{a}'^{(S2)}$ to the interval $(0, 1)$. These steps are functionally motivated and not by reference to biology.

We first apply a size limit:

$$\mathbf{a}_j''^{(S2)}[t] = \begin{cases} \mathbf{a}_j'^{(S2)}[t], & \text{if } \mathbf{a}_j'^{(S2)}[t] \leq \lambda \\ \lambda - \frac{1}{2}(\mathbf{a}_j'^{(S2)}[t] - \lambda), & \text{otherwise.} \end{cases} \quad (9)$$

According to this formula, \mathbf{a}'' grows undiminished with \mathbf{a}' until it reaches the value λ , and thereafter actually diminishes with slope $-1/2$. (In our experiments we found the value $\lambda = 1.3 \cdot \frac{h^{(W2)} + w^{(W2)}}{2}$ to work well). This saturation helps to mitigate imbalances between network fragments with differing numbers of participating neurons (as with our stimuli, line segments, where many more feature neurons activate around the end-points than in the middle of segments).

In the next step, we introduce the normalization

$$\mathbf{a}_{c,i}^{\text{norm}}{}^{(S2)}[t] = \frac{\mathbf{a}_{c,i}''^{(S2)}[t]}{\max_{i'} \mathbf{a}_{c,i'}''^{(S2)}[t]} \quad (10)$$

This normalization confines activations to the interval $(0, 1)$, where neurons receiving maximal support are mapped to 1, while those with lesser support are mapped to correspondingly smaller values. This normalization relative to the maximal value that itself is growing during learning instead of relative to a constant parameter like λ is important to give neurons initially (when lateral connections are still small) a chance to fire and later, when lateral links of often co-active neurons have grown, to suppress those neurons that are part of accidental or noise patterns.

E Alternative Neuron Selection

Only one of a set of alternative neurons is allowed to be active. The selection process is determined by evaluating the correlation between feature patches and alternative kernel filters. Initially, patches matching the size of the feature kernel are extracted from every position in the input $(\mathbf{y}^{(S1)}; \mathbf{y}^{(S2)}[t-1])$. Subsequently, the correlation of these patches with the feature kernels is computed, and at each location, the alternative kernel exhibiting the highest correlation is chosen. Consequently, multiple features may be active at each location, but only one of the competing feature channels — the one best matching a given input patch — achieves activation. At initialization (see Appendix B), where all alternative channels fit equally well to a given local pattern, the first channel is selected. However, after applying a Hebbian update, this channel starts specializing in recognizing a specific local pattern. Consequently, the activation of subsequent local patterns may either trigger the same channel if they have a high correlation or activate another channel better suited to recognizing the new pattern.

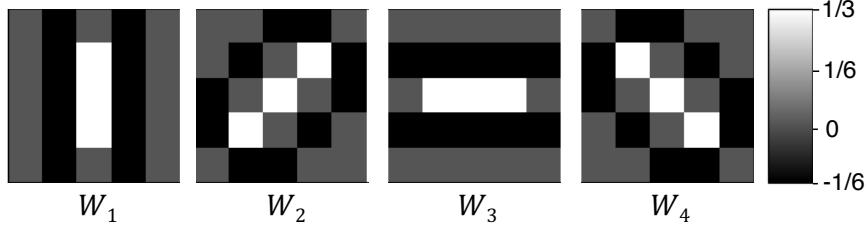


Figure 9: The hand-crafted convolutional kernels of the feature extractor.

F Feature Extraction

The input images have $C = 1$ channel and $C^{(S1)} = 4$ features are extracted at each spatial location. We utilize hand-crafted filters with a size of 5×5 as displayed in Figure 9, each of the four filters corresponding to a particular line orientation (vertical, positive diagonal, horizontal, and negative diagonal). This approach is motivated by the observation that, given our specific elementary feature types (cf. Fig. 9), straight lines, when viewed locally, can be expressed as combinations of these fundamental line types (e.g., a line with a 20° angle activates the horizontal and the $+45^\circ$ diagonal feature neurons). The use of such manually designed filters allows a more straightforward interpretation of the results. While more complex datasets necessitate learned filters, our primary focus in this work is on constructing net fragments, and thus, we favor the simplicity of hand-crafted filters.

These filters move across the image with a step size (“stride”) of 1 (assuming 0 as input signal beyond the image border: “zero padding”) so that the input image and output image are of the same size. Thus, the input \mathbf{x} in the feature extraction stage are images of size $(1 \times 32 \times 32)$ and the output $\mathbf{y}^{(S1)}$ feature activation maps of size $(4 \times 32 \times 32)$. The dataset’s lines activate multiple filters at different positions, facilitating the construction of net fragments across these feature channels. We convert the floating-point output of the convolutional weights to binary activations using binary neurons $B()$ with a bias of $b^{(S1)} = 0.5$, i.e. neurons fire a binary spike if $a_j > 0.5$ (see eq. 2).

G Parameters

We train the model with a learning rate of $\alpha = 0.2$ for 100 training cycles (epochs). The model uses $n_a = 10$ alternative neurons with lateral connections spanning a kernel of 11×11 neurons. Since we use $C^{(S1)} = 4$ filters for feature extraction in $S1$ (see Appendix F), the number of alternative channels in $S2$ corresponds to $C^{(S2)} = 4 \cdot n_a = 40$. The resulting weight matrices have dimensions $\mathbf{W}^{(F)} \in \mathbb{R}^{40 \times 4 \times 11 \times 11}$ and $\mathbf{W}^{(L)} \in \mathbb{R}^{40 \times 40 \times 11 \times 11}$, respectively $\mathbf{W}^{(S2)} \in \mathbb{R}^{40 \times 44 \times 11 \times 11}$.

H Learned Support Strength

A crucial parameter for gauging the efficacy of net fragments involves quantifying the difference in support received by active and inactive neurons. During the initial stages of training, active neurons receive a support of 1, and inactive neurons receive a support of 0. The support increases during training as net fragments are formed. Figure 10 visually represents the average support for active and inactive neurons, captured by a'_j (see eqs. 3, 5). Notably, the graph demonstrates a widening gap between the support for active and inactive neurons. This discrepancy suggests an enhancement in the robustness of statistically relevant patterns over time as active neurons receive increased support and the support of inactive neurons remains close to 0.

I Quantifying Noise Filtration

Our evaluations focus on the efficacy of net fragments in suppressing added Gaussian noise and complementing missing or occluded patterns. For both evaluation experiments, we report *recall* and *precision*: Recall is defined as the proportion of neurons that maintain activity (on) when noise is added, and precision quantifies the extent to which neurons activated in the presence of noise were also active without noise. To assess the system’s noise reduction capability in the case of Gaussian noise, we additionally measure the percentage of flipped neurons that revert to their original activation state, referring to this measurement as the *noise reduction rate*. To assess subtractive noise, we measure the similarity

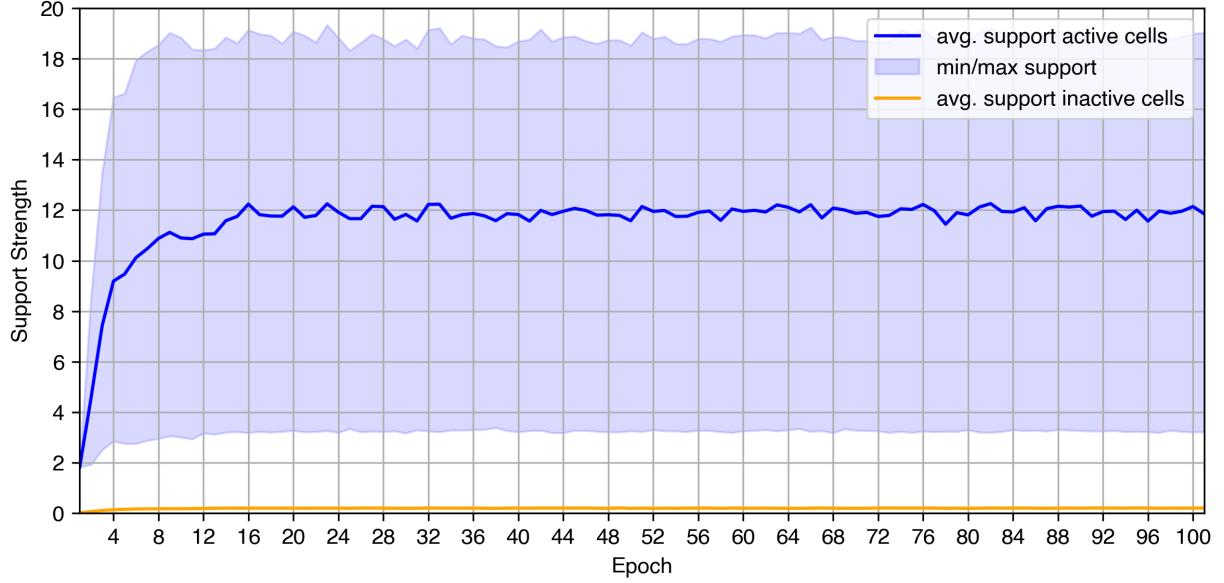


Figure 10: The average support active and inactive neurons receive during training, captured by a'_j . The support strength is shown on the y-axis, and the training progress (epoch) is on the x-axis.

between the original (created without noise) and reconstructed (created with noise) feature activations at the spatial locations where pixels have been removed. We call this metric the *feature reconstruction rate*.

Filtering Gaussian Noise. Figure 11 displays noise reduction rates, recall, and precision across different parameter configurations for different levels of added noise. The metrics demonstrate that increasing the inhibition coefficient or the activation bias results in enhanced noise filtration.

The overall efficacy of noise reduction remains consistently high across most parameter configurations, exhibiting a constant noise reduction rate of $\geq \approx 95\%$. This rate persists regardless of the magnitude of the introduced noise. As the noise suppression rate is approximately constant, this means that if there is more noise in the input, there will also be more noise in the output, albeit limited to around 5% of the input noise.

Reconstructing Subtractive Noise. Figure 12 displays the feature reconstruction rate, precision, and recall for removed line fragments of different lengths. Having a lower activation bias or inhibition coefficient γ leads to a better reconstruction of removed patterns, as these settings allow neurons with less support to activate.

Interpreting Metrics. While the reported values for precision and recall may appear low, we argue that these fragments are of good quality and point out that precision and recall metrics have to be interpreted differently than in a classification context: A net fragment consisting of many neurons characterizes (a part of) an object and maintains this characterization even when many of its neurons are inverted (and precision and recall are low). This aligns with the findings of Ahmad and Hawkins (2015), highlighting that binary distributed activations demonstrate high robustness, retaining accurate interpretation even in the presence of numerous flipped neurons.

In the case of the depicted straight lines, the feature neuron activations of the two most similar lines overlap with 21.2%. Therefore, achieving precision and recall values surpassing this threshold permits reliable discrimination even among similar lines. Hence, we posit that various of the tested parameter configurations are well suited as they permit this distinction.

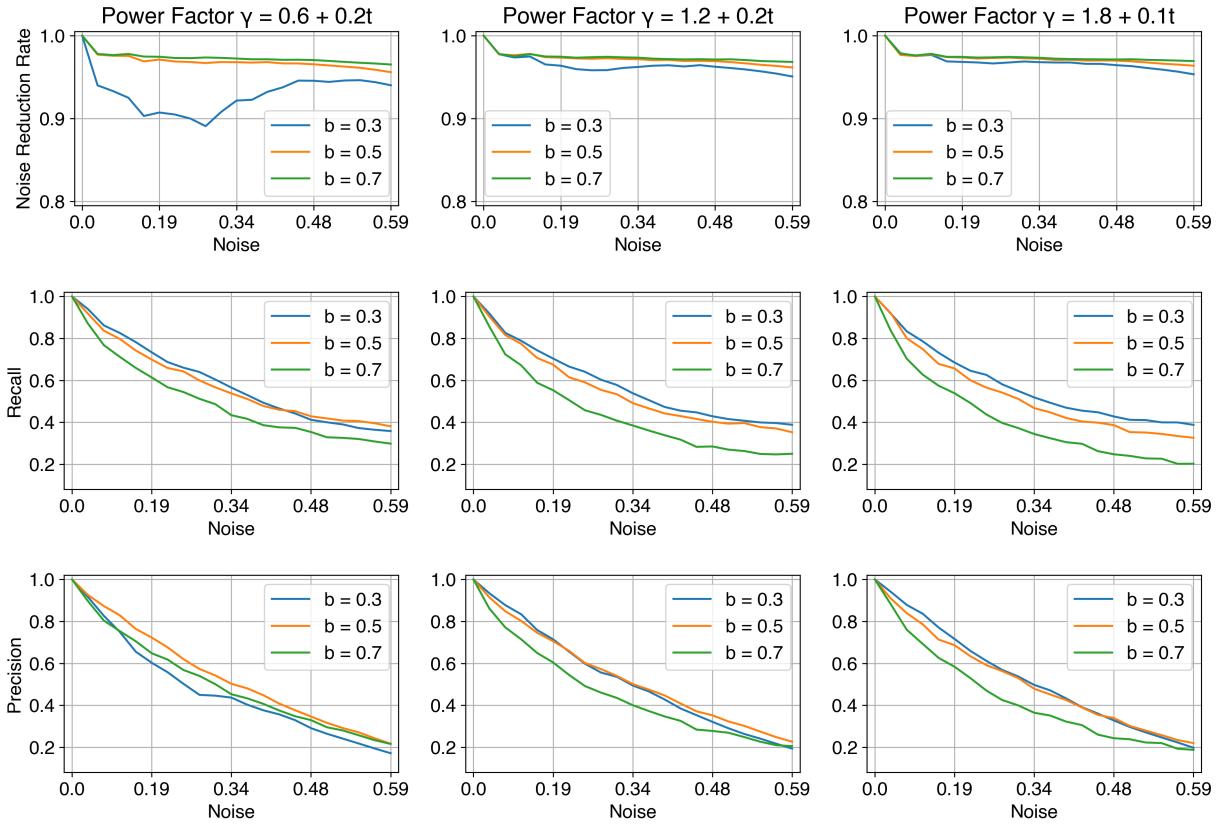


Figure 11: System performance as function of the noise level. First row: noise reduction rate; second row: recall; third row: precision. Columns differ in the inhibition coefficient γ that models inhibition. The colors within each plot represent different activation biases $b^{(S2)}$.

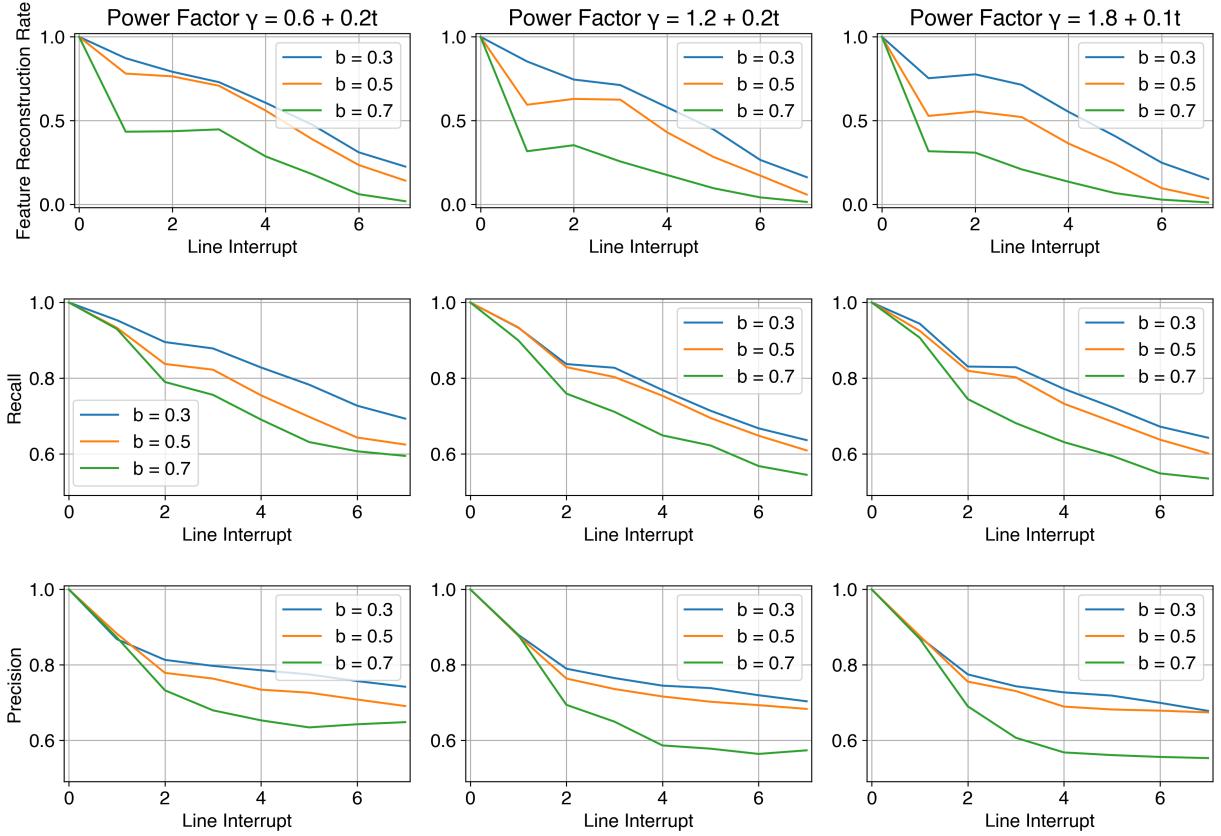


Figure 12: Reconstruction of partial occlusion. *First row:* feature reconstruction rate; *second row:* recall; *third row:* precision. Columns are distinguished by the inhibition coefficient γ , and line colors within plots distinguish activation bias $b^{(S2)}$.

References

- Abbott, L. F. and Nelson, S. B. (2000). Synaptic plasticity: taming the beast. *Nature Neuroscience*, 3(S11):1178–1183.
- Ahmad, S. and Hawkins, J. (2015). Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory. *arXiv*, (preprint arXiv:1503.07469).
- Ahmad, S. and Scheinkman, L. (2019). How Can We Be So Dense? The Benefits of Using Highly Sparse Representations. *arXiv*, (preprint arXiv:1903.11257).
- Amirian, M., Schwenker, F., and Stadelmann, T. (2018). *Trace and Detect Adversarial Attacks on CNNs Using Feature Response Maps*, volume 11081 of *Lecture Notes in Computer Science*, page 346–358.
- Anderson, C. H. and van Essen, D. C. (1987). Shifter circuits: a computational strategy for dynamic aspects of visual processing. *Proceedings of the National Academy of Sciences*, 84(17):6297–6301.
- Bhagoji, A. N., Cullina, D., Sitawarin, C., et al. (2018). Enhancing robustness of machine learning systems via data transformations. In *Proceedings of the 52nd Annual Conference on Information Sciences and Systems*, pages 1–5.
- Biederman, I. and Kalocsai, P. (1997). Neurocomputational bases of object and face recognition. *Philosophical Transactions of the Royal Society*, 352(1358):1203–1219.
- Brendel, W., Rauber, J., and Bethge, M. (2018). Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. *arXiv*, (preprint arXiv:1712.04248).
- Cao, K., Liu, M., Su, H., Wu, J., Zhu, J., and Liu, S. (2021). Analyzing the noise robustness of deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3289–3304.
- Carlini, N. and Wagner, D. Towards Evaluating the Robustness of Neural Networks. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, pages 39–57.
- Carlucci, F. M., Russo, P., Tommasi, T., et al. (2019). Hallucinating Agnostic Images to Generalize Across Domains. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop*, pages 3227–3234.
- Chen, P.-Y., Zhang, H., Sharma, Y., et al. (2017). ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26.
- Cisse, M., Adi, Y., Neverova, N., et al. (2017). Houdini: Fooling Deep Structured Prediction Models. *arXiv*, (preprint arXiv:1707.05373).
- Cohen, M. A. and Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, (5):815–826.
- Corchado, E., Han, Y., and Fyfe, C. (2003). Structuring global responses of local filters using lateral connections. *Journal of Experimental & Theoretical Artificial Intelligence*, 15(4):473–487.
- Deriu, J., Tuggener, D., Von Däniken, P., et al. (2022). Probing the Robustness of Trained Metrics for Conversational Dialogue Systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 750–761.
- Dou, Q., Castro, D. C., Kamnitsas, K., et al. (2019). Domain Generalization via Model-Agnostic Learning of Semantic Features. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, volume 32, pages 6450–6461.
- Drenkow, N., Sani, N., Shpitser, I., et al. (2022). A Systematic Review of Robustness in Deep Learning for Computer Vision: Mind the gap? *arXiv*, (preprint arXiv:2112.00639).
- Du, Y., Xu, J., Xiong, H., et al. (2020). Learning to Learn with Variational Information Bottleneck for Domain Generalization. In *European Conference on Computer Vision*, volume 12355, pages 200–216.
- Dziugaite, G. K., Ghahramani, Z., and Roy, D. M. (2016). A study of the effect of JPG compression on adversarial images. *arXiv*, (preprint arXiv:1608.00853).
- Engstrom, L., Tran, B., Tsipras, D., et al. (2019). Exploring the Landscape of Spatial Robustness. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 1802–1811.
- Erfani, S. M., Baktashmotlagh, M., Moshtaghi, M., et al. (2016). Robust Domain Generalisation by Enforcing Distribution Invariance. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 1455–1461.
- Fernandes, T. and von der Malsburg, C. (2015). Self-Organization of Control Circuits for Invariant Fiber Projections. *Neural Computation*, 27(5):1005–1032.

- Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3 – 71.
- Forrest, S. (1990). Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks: Introduction to the proceedings of the ninth annual cnls conference. *Physica D: Nonlinear Phenomena*, 42(1):1–11.
- French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3:128–135.
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.
- Gao, J., Wang, B., Lin, Z., et al. (2017). DeepCloak: Masking Deep Neural Network Models for Robustness Against Adversarial Samples. *arXiv*, (preprint arXiv:1702.06763).
- Ghifary, M., Balduzzi, D., Kleijn, W. B., et al. (2017). Scatter Component Analysis: A Unified Framework for Domain Adaptation and Domain Generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(7):1414–1430.
- Gilbert, C., Hirsch, J., and Wiesel, T. (1990). Lateral Interactions in Visual Cortex. *Cold Spring Harbor Symposia on Quantitative Biology*, 55(1):663–677.
- Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *arXiv*, (preprint arXiv:1412.6572).
- Grill-Spector, K. and Malach, R. (2004). The human visual cortex. *Annual Review of Neuroscience*, 27(1):649–677.
- Guo, Y., Zhang, C., Zhang, C., et al. (2018). Sparse DNNs with Improved Adversarial Robustness. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, volume 31, pages 240–249.
- Hebb, D. O. (1949). *The Organization of Behavior; A Neuropsychological Theory*.
- Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558.
- Ilyas, A., Santurkar, S., Tsipras, D., et al. (2019). Adversarial Examples Are Not Bugs, They Are Features. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, volume 32, pages 125–136.
- Ito, M., Tamura, H., Fujita, I., et al. (1995). Size and position invariance of neuronal responses in monkey inferotemporal cortex. *Journal of Neurophysiology*, 73(1):218 –226.
- Keller, G. B. and Mrsic-Flogel, T. D. (2018). Predictive processing: A canonical cortical computation. *Neuron*, 100(2):424–435.
- Köhler, W. (1992). *Gestalt psychology: an introduction to new concepts in modern psychology*.
- Kothari, R. and Agyepong, K. (1996). On lateral connections in feed-forward neural networks. In *Proceedings of International Conference on Neural Networks*, volume 1, pages 13–18.
- Lades, M., Vorbruggen, J. C., Buhmann, J., et al. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- LeCun, Y., Boser, B., Denker, J. S., et al. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551.
- Leuba, G. and Kraftsik, R. (1994). Changes in volume, surface estimate, three-dimensional shape and total number of neurons of the human primary visual cortex from midgestation until old age. *Anat Embryol*, 190:351–366.
- Li, P., Yi, J., Zhou, B., et al. (2019). Improving the Robustness of Deep Neural Networks via Adversarial Training with Triplet Loss. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2909–2915.
- Liao, N., Wang, S., Xiang, L., et al. (2022). Achieving adversarial robustness via sparsity. *Machine Learning*, 111(2):685–711.
- Linnainmaa, S. (1970). *The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors*. PhD thesis, Master's Thesis (in Finnish), Univ. Helsinki.
- Luz, Y. and Shamir, M. (2012). Balancing Feed-Forward Excitation and Inhibition via Hebbian Inhibitory Synaptic Plasticity. *PLoS Computational Biology*, 8(1):e1002334.

- Marr, D. (2010). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*.
- Meng, D. and Chen, H. (2017). MagNet: A Two-Pronged Defense against Adversarial Examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147.
- Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., et al. (2017). Universal Adversarial Perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 86–94.
- Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582.
- Neururer, D., Dellwo, V., and Stadelmann, T. (2024). Deep neural networks for automatic speaker recognition do not learn supra-segmental temporal features. *Pattern Recognition Letters*, 181:64–69.
- Olshausen, B. A., Anderson, C. H., and Van Essen, D. C. (1993). A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13(11):4700–4719.
- Papadimitriou, C., Vempala, S., Mitropolsky, D., et al. (2020). Brain computation by assemblies of neurons. *Proceedings of the National Academy of Sciences*, 117(25):14464–14472.
- Papernot, N., McDaniel, P., Jha, S., et al. (2016a). The Limitations of Deep Learning in Adversarial Settings. In *Proceedings of the IEEE European Symposium on Security and Privacy*, pages 372–387.
- Papernot, N., McDaniel, P., Wu, X., et al. (2016b). Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 582–597.
- Prince, S. J. D. (2023). *Understanding Deep Learning*.
- Rahman, M. M., Fookes, C., Baktashmotagh, M., et al. (2020). Correlation-aware adversarial domain adaptation and generalization. *Pattern Recognition*, 100:107124.
- Ros, A. S. and Doshi-Velez, F. (2018). Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, volume 203, pages 1660–1669.
- Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*.
- Rumelhart, D. E. and McClelland, J. L. (1987). *Learning Internal Representations by Error Propagation*, pages 318–362.
- Rusak, E., Schott, L., Zimmermann, R. S., et al. (2020). A Simple Way to Make Neural Networks Robust Against Diverse Image Corruptions. In *Proceedings of the 16th European Conference on Computer Vision*, volume 12348, pages 53–69.
- Sager, P., Salzmann, S., Burn, F., et al. (2022). Unsupervised Domain Adaptation for Vertebrae Detection and Identification in 3D CT Volumes Using a Domain Sanity Loss. *Journal of Imaging*, 8(8):222.
- Sarkar, S., Bansal, A., Mahbub, U., et al. (2017). UPSET and ANGRI : Breaking High Performance Image Classifiers. *arXiv*, (preprint arXiv:1707.01159).
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.
- Simmler, N., Sager, P., Andermatt, P., et al. (2021). A Survey of Un-, Weakly-, and Semi-Supervised Learning Methods for Noisy, Missing and Partial Labels in Industrial Vision Applications. In *Proceedings of the 8th Swiss Conference on Data Science*, pages 26–31.
- Stadelmann, T., Amirian, M., Arabaci, I., Arnold, M., Duivesteijn, G. F., Elezi, I., Geiger, M., Lörväld, S., Meier, B. B., Rombach, K., and Tuggener, L. (2018). Deep learning in the wild. In *Proceedings of 8th International Association for Pattern Recognition TC3 Workshop*, pages 17–38.
- Stadelmann, T., Tolkachev, V., Sick, B., Stampfli, J., and Dürr, O. (2019). Beyond imagenet: deep learning in industrial practice. *Applied data science: lessons learned for the data-driven business*, pages 205–232.
- Stettler, D. D., Das, A., Bennett, J., et al. (2002). Lateral Connectivity and Contextual Interactions in Macaque Primary Visual Cortex. *Neuron*, 36(4):739–750.
- Suzuki, M., Pennartz, C. M. A., and Aru, J. (2023). How deep is the brain? the shallow brain hypothesis. *Nature Reviews Neuroscience*, 24(12):778–791.
- Szegedy, C., Zaremba, W., Sutskever, I., et al. (2014). Intriguing properties of neural networks. *arXiv*, (preprint arXiv:1312.6199).

- Taubman, D. and Marcellin, M. (2002). *Jpeg-2000 image compression: fundamentals, standards and practice*. Kluwer Academic Publishers, Dordrecht.
- Timpl, L., Entezari, R., Sedghi, H., et al. (2022). Understanding the effect of sparsity on neural networks robustness. *arXiv*, (preprint arXiv:2206.10915).
- Tuggener, L., Emberger, R., Ghosh, A., et al. (2024). Real World Music Object Recognition. *Transactions of the International Society for Music Information Retrieval*, 7(1):1–14.
- Tuggener, L., Schmidhuber, J., and Stadelmann, T. (2022). Is it enough to optimize cnn architectures on imagenet? *Frontiers in Computer Science*, 4:1041703.
- Usrey, W. M., Sceniak, M. P., and Chapman, B. (2003). Receptive fields and response properties of neurons in layer 4 of ferret visual cortex. *Journal of Neurophysiology*, 89(2):1003–1015.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is All you Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, page 6000–6010.
- Vogels, T. P., Sprekeler, H., Zenke, F., et al. (2011). Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks. *Science*, 334(6062):1569–1573.
- von der Malsburg, C. (2014). A Vision Architecture. *arXiv*, (preprint arXiv:1407.1642).
- von der Malsburg, C. (2018). Concerning the Neuronal Code. *Journal of Cognitive Science*, 19(4):511–550.
- von der Malsburg, C. (2023). Fodor and pelyshyn’s critique of connectionism and the brain as basis of the mind.
- von der Malsburg, C., Grewe, B. F., and Stadelmann, T. (2022a). Making sense of the natural environment. In *The Biannual Conference of the German Cognitive Science Society*.
- von der Malsburg, C., Stadelmann, T., and Grewe, B. F. (2022b). A Theory of Natural Intelligence. *arXiv*, (preprint arXiv:2205.00002).
- Waibel, A., Hanazawa, T., Hinton, G. E., et al. (1987). Phoneme Recognition Using Time-Delay Neural Networks. In *Meeting of the Institute of Electrical*, volume 37, pages 329–339.
- Wallace, G. K. (1991). The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44.
- Wang, H., Ge, S., Lipton, Z., et al. (2019). Learning Robust Global Representations by Penalizing Local Predictive Power. In *Proceedings of the 33th International Conference on Neural Information Processing Systems*, volume 32, pages 10506–10518.
- Wertheimer, M., Köhler, W., Fuchs, W., et al. (1938). *A source book of Gestalt psychology*.
- Windrim, L., Melkumyan, A., Murphy, R., et al. (2016). Unsupervised feature learning for illumination robustness. In *Proceedings of the IEEE International Conference on Image Processing 2016*, pages 4453–4457.
- Wolfrum, P., Wolff, C., Lücke, J., et al. (2008). A recurrent dynamic model for correspondence-based face recognition. *Journal of Vision*, 8(7):34.
- Xu, W., Evans, D., and Qi, Y. (2018). Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *Proceedings of the Network and Distributed System Security Symposium*.
- Yan, P., Abdulkadir, A., Luley, P.-P., Rosenthal, M., Schatte, G. A., Grewe, B. F., and Stadelmann, T. (2024). A comprehensive survey of deep transfer learning for anomaly detection in industrial time series: Methods, applications, and directions. *IEEE Access*, 12:3768–3789.
- Zhang, L., Wang, X., Yang, D., et al. (2020). Generalizing Deep Learning for Medical Image Segmentation to Unseen Domains via Deep Stacked Transformation. *IEEE Transactions on Medical Imaging*, 39(7):2531–2540.
- Zhou, K., Yang, Y., Hospedales, T., et al. (2020). Learning to Generate Novel Domains for Domain Generalization. In *European Conference on Computer Vision*, volume 12361, pages 561–578.
- Zhu, J. and von der Malsburg, C. (2004). Maplets for correspondence-based object recognition. *Neural Networks*, 17(8):1311–1326.