

MBZUAI

Digital.Commons@MBZUAI

Computer Vision Faculty Publications

Scholarly Works

4-16-2022

Visual Attention Methods in Deep Learning: An In-Depth Survey

Mohammed Hassanin

UniSA (University of South Australia), Faculty of Computers and Information, Fayoum University, Egypt

Anwar Saeed

The Australian National University, Canberra, Australia & Data61, CSIRO (The Commonwealth Scientific and Industrial Research Organisation), The University of Technology Sydney, University of Canberra, Australia

Ibrahim Radwan

University of Canberra, Australia

Fahad Shahbaz Khan

Mohamed bin Zayed University of Artificial Intelligence & Linköping University, Sweden

Ajmal Mian

The University of Western Australia, Australia

Follow this and additional works at: <https://dclibrary.mbzuai.ac.ae/cvfp>



Part of the [Artificial Intelligence and Robotics Commons](#)

Preprint: arXiv

Archived with thanks to arXiv

Preprint License: CC by 4.0

Uploaded 24 May 2022

Recommended Citation

M. Hassanin, A. Saeed, I. Radwan, F.S. Khan, and A. Mian, "Visual Attention Methods in Deep Learning: An In-Depth Survey", arXiv, Apr 2022, doi: 10.48550/arXiv.2204.07756

This Article is brought to you for free and open access by the Scholarly Works at Digital.Commons@MBZUAI. It has been accepted for inclusion in Computer Vision Faculty Publications by an authorized administrator of Digital.Commons@MBZUAI. For more information, please contact libraryservices@mbzuai.ac.ae.

Visual Attention Methods in Deep Learning: An In-Depth Survey

Mohammed Hassanin, Saeed Anwar, Ibrahim Radwan, Fahad S Khan and Ajmal Mian

Abstract—Inspired by the human cognitive system, attention is a mechanism that imitates the human cognitive awareness about specific information, amplifying critical details to focus more on the essential aspects of data. Deep learning has employed attention to boost performance for many applications. Interestingly, the same attention design can suit processing different data modalities and can easily be incorporated into large networks. Furthermore, multiple complementary attention mechanisms can be incorporated in one network. Hence, attention techniques have become extremely attractive. However, the literature lacks a comprehensive survey specific to attention techniques to guide researchers in employing attention in their deep models. Note that, besides being demanding in terms of training data and computational resources, transformers only cover a single category in self-attention out of the many categories available. We fill this gap and provide an in-depth survey of 50 attention techniques categorizing them by their most prominent features. We initiate our discussion by introducing the fundamental concepts behind the success of attention mechanism. Next, we furnish some essentials such as the strengths and limitations of each attention category, describe their fundamental building blocks, basic formulations with primary usage, and applications specifically for computer vision. We also discuss the challenges and open questions related to attention mechanism in general. Finally, we recommend possible future research directions for deep attention.

Index Terms—Attention Mechanisms, Deep Attention, Attention Modules, Attention in Computer Vision and Machine learning.

1 INTRODUCTION

ATTENTION has a natural bond with the human cognitive system. According to cognitive science, the human optic nerve receives massive amounts of data, more than it can process. Thus, the human brain weighs the input and pays attention only to the necessary information. With recent developments in machine learning, more specifically, deep learning, and the increasing ability to process large and multiple input data streams, researchers have adopted a similar concept in many domains and formulated various attention mechanisms to improve the performance of deep neural network models in machine translation [1], [2], visual recognition [3], generative models [4], multi-agent reinforcement learning [5], *etc.* Over the past decade, deep learning has advanced in leaps and bounds, leading to many deep neural network architectures capable of learning complex relationships in data. Generally, neural networks provide implicit attention to extract meaningful information from the data.

Explicit attention mechanism in deep learning was first introduced to tackle the *forgetting* issue in encoder-decoder architectures designed for the machine translation problem [6]. Since the network's encoder part focuses on generating a representative input vector, the decoder generates the output from the representation vector. A bi-directional Recurrent Neural Network (RNN) [6] is employed for solving the *forgetting* issue by generating a context vector from the input sequence and then decoding the output based on the context vector as well as the previous hidden states. The context vector is computed by a weighted sum of the intermediate representations which makes this method an example of explicit attention. Moreover, Long-Short-Term-Memory (LSTM) [7] is employed to generate both the context vector and the output. Both methods compute the context vector considering all the hidden states of the encoder. However, [8] introduced another idea by getting the attention mechanism to focus on only a subset of the hidden states to generate every item in the context vector. This was computationally less expensive compared to the previous attention methods and shows a trade-off between *global* and *local* attention mechanisms.

Another attention-based breakthrough was made by Vaswani *et al.* [2], where an entire architecture was created based on the self-attention mechanism. The items in the input sequence are first encoded in parallel into multiple representations called key, query, and value. This architecture, coined the Transformer, helps capture the importance of each item relative to others in the input sequence more effectively. Recently, many researchers have extended the basic Transformer architecture for specific applications.

To pay attention to the significant parts in an image and suppress unnecessary information, advancements of attention-based learning have found their way into mul-

- Mohammed Hassanin is with UniSA (University of South Australia), and Faculty of Computers and Information, Fayoum University, Egypt. E-mail: mohammed.hassanin@unisa.edu.au.
- Saeed Anwar is with College of Engineering and Computer Science, The Australian National University, Canberra, Australia. He is also affiliated with Data61, CSIRO (The Commonwealth Scientific and Industrial Research Organisation), The University of Technology Sydney, and University of Canberra, Australia.
- Ibrahim Radwan is with University of Canberra, Australia.
- Fahad S. Khan is an Associate Professor with Mohammad Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE, and Computer Vision Laboratory, Linköping University, Sweden.
- Ajmal Mian is a Professor of Computer Science with The University of Western Australia, Australia.
- Corresponding Author: Saeed Anwar (E-mail: saeed.anwar@anu.edu.au)

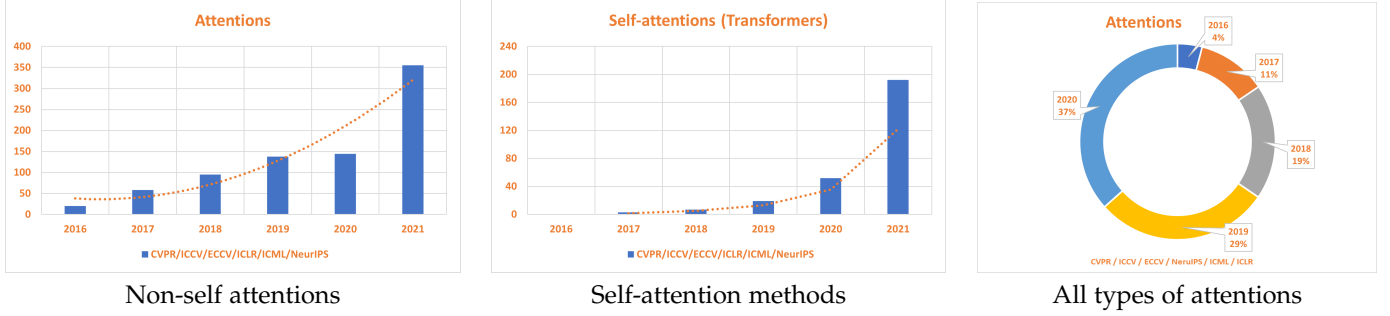


Fig. 1. Visual charts show the increase in the number of attention related papers in the top conferences including CVPR, ICCV, ECCV, NeurIPS, ICML, and ICLR.

multiple computer vision tasks, either employing a different attention map for every image pixel, comparing it with the representations of other pixels [3], [4], [9] or generating an attention map to extract the global representation for the whole image [10], [11]. However, the design of attention mechanism is highly dependent on the problem at hand. To enforce the selection of hidden states that correspond to the critical information in the input, attention techniques have been used as plug-in units in vision-based tasks, alleviating the risk of vanishing gradients. To sum up, attention scores are calculated, and hidden states are selected either deterministically or stochastically.

Attention has been the center of significant research efforts over the past few years and image attention has been flourishing in many different machine learning and vision applications, for example, classification [12], detection [13], image captioning [14], 3D analysis [15], *etc.* Despite the impressive performance of attention techniques employed in deep learning, there is no literature survey that comprehensively reviews all, especially deep learning based, attention mechanisms in vision to categorize them based on their basic underlying architectures and highlight their strengths and weaknesses. Recently, researchers surveyed application-specific attention techniques with emphasis on NLP-based [16], transformer-based [17], [18], and graph-based approaches [19]. However, no comprehensive study collates with the huge and diverse scope of *all* deep learning based attention techniques developed for visual inputs.

In this article, we review attention techniques specific to vision. Our survey covers the numerous basic building blocks (operations and functions) and complete architectures designed to learn suitable representations while making the models attentive to the relevant and important information in the input images or videos. Our survey broadly classifies attention mechanisms proposed in the computer vision literature including soft attention, hard attention, multi-modal, arithmetic, class attention, and logical attention. We note that some methods belong to more than one category; however, we assign each method to the category where it has a dominant association with other methods of that category. Following such a categorization helps track the common attention mechanism characteristics and offers insights that can potentially help in designing novel attention techniques. Figure 2 shows the classification of the attention mechanisms.

We emphasize that a survey is warranted for attention

in vision due to the large number of papers published as outlined in Figure 1. It is evident from Figure 1 that the number of articles published in the last year has significantly increased compared to previous years, and we expect to see a similar trend in the coming years. Furthermore, our survey lists articles of significant importance to assist the computer vision and machine learning community in adopting the most suitable attention mechanisms in their models and avoid duplicating attention methodologies. It also identifies research gaps, provides the current research context, presents plausible research directions and future areas of focus.

Since transformers have been employed across many vision applications; a few surveys [17], [18] summarize the recent trends of transformers in computer vision. Although transformers offer high accuracy, this comes at the cost of very high computational complexity which hinders their feasibility for mobile and embedded system applications. Furthermore, transformer based models require substantially more training data than CNNs and lack efficient hardware designs and generalizability. According to our survey, transformers only cover a single category in self-attention out of the 50 different attention categories surveyed. Another significant difference is that our survey focuses on attention types rather than applications covered in transformer-based surveys [17], [18].

2 ATTENTION IN VISION

The primary purpose of the attention in vision is to imitate the human visual cognitive system and focus on the essential features [20] in the input image. We categorize attention methods based on the main function used to generate attention scores, such as softmax or sigmoid. Table 1 provides the summary, application, strengths, and limitations for the category presented in this survey.

2.1 Soft (Deterministic) Attention

This section reviews soft-attention methods such as channel attention, spatial attention, and self-attention. In channel attention, the scores are calculated channel wise because each one of the feature maps (channels) attend to specific parts of the input. In spatial attention, the main idea is to attend to the critical regions in the image. Attending over regions of interest facilitates object detection, semantic segmentation, and person re-identification. In contrast to channel

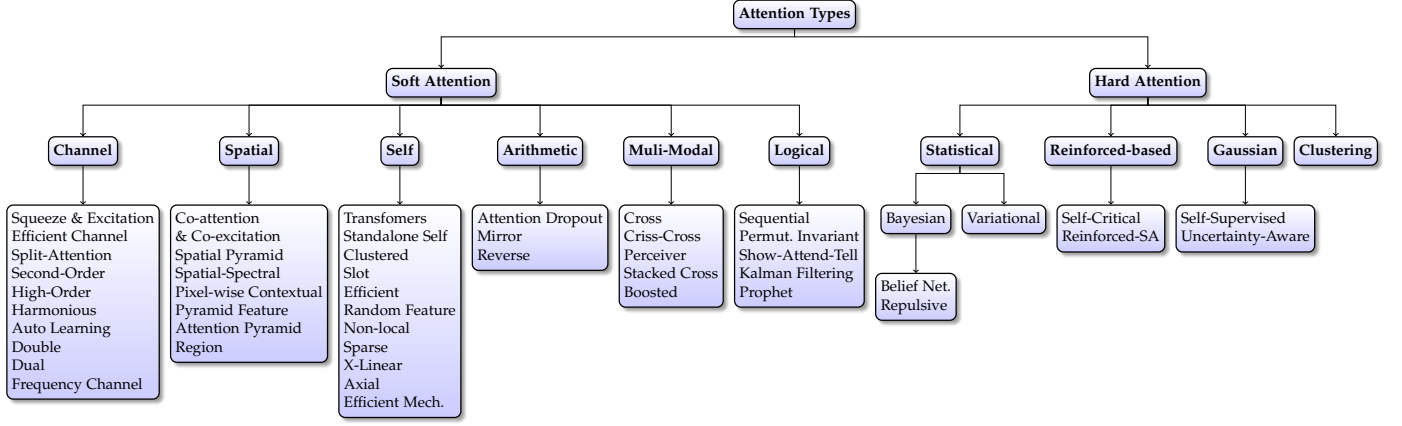


Fig. 2. A taxonomy of attention types. The attentions are categorized based on the methodology adopted to perform attention. Some of the attention techniques can be accommodated in multiple categories; in this case, the attention is grouped based on the most dominant characteristic and primary application.

attention, spatial attention attends to the important parts in the spatial map (bounded by width and height). It can be used independently or as a complementary mechanism to channel attention. On the other hand, self-attention is proposed to encode higher-order interactions and contextual information by extracting the relationships between input sequence tokens. It is different from channel attention in how it generates the attention scores, as it mainly calculates the similarity between two maps (K, Q) of the same input, whereas channel attention generates the scores from a single map. However, self attention and channel attention both operate on channels. Soft attention methods calculate the attention scores as the weighted sum of all the input entities [8] and mainly use soft functions such as softmax and sigmoid. Since these methods are differentiable, they can be trained through back-propagation techniques. However, they suffer from other issues such as high computational complexity and assigning weights to non-attended objects.

2.1.1 Channel Attention

Squeeze & Excitation Attention: The Squeeze-and-Excitation (SE) Block [21], shown in Figure 3(a), is a unit design to perform dynamic channel-wise feature attention. The SE attention takes the output of a convolution block and converts each channel to a single value via global average pooling; this process is called “squeeze”. The output channel ratio is reduced after passing through the fully connected layer and ReLU for adding non-linearity. The features are passed through the fully connected layer, followed by a sigmoid function to achieve a smooth gating operation. The convolutional block feature maps are weighted based on the side network’s output, called the “excitation”. The process can be summarized as

$$f_s = \sigma(FC(ReLU(FC(f_g)))), \quad (1)$$

where FC is the fully connected layer, f_g is the average global pooling, σ is the sigmoid operation. The main intuition is to choose the best representation of each channel in order to generate attention scores.

Efficient Channel Attention (ECA) [22] is based on squeeze & excitation network [21] and aims to increase efficiency as well as decrease model complexity by removing the dimensionality reduction. ECA (see Fig 3(g)) achieves cross-channel interaction locally through analyzing each channel and its k neighbors, following channel-wise global average pooling but with no dimensionality reduction. ECA accomplishes efficient processing via fast 1D convolutions. The size k represents the number of neighbors that can participate in one channel attention prediction *i.e.* the coverage of local cross-channel interaction.

Split-Attention Networks: ResNest [23], a variant of ResNet [24], uses split attention blocks as shown in Figure 3(h). Attention is obtained by summing the inputs from previous modules and applying global pooling, passing through a composite function *i.e.* convolutional layer-batch normalization-ReLU activation. The output is again passed through convolutional layers. Afterwards, a softmax is applied to normalize the values and then multiply them with the corresponding inputs. Finally, all the features are summed together. This mechanism is similar to the squeeze & excitation attention [21]. ResNest is also a special type of squeeze & excitation where it squeezes the channels using average pooling and summing of the split channels.

Channel Attention in CBAM: Convolutional Block Attention Module (CBAM) [25] employs channel attention and exploits the inter-channel feature relationship as each feature map channel is considered a feature detector focusing on the “what” part of the input image. The input feature map’s spatial dimensions are squeezed for computing the channel attention followed by aggregation while using both average-pooling and max-pooling to obtain two descriptors. These descriptors are forwarded to a three-layer shared multi-layer perceptron (MLP) to generate the attention map. Subsequently, the output of each MLP is summed element-wise and then passed through a sigmoid function as shown in Figure 3(b). In summary, the channel attention is computed as

$$f_{ch} = \sigma(MLP(MaxPool(f)) + MLP(AvgPool(f))), \quad (2)$$

where σ denotes the sigmoid function, and f represents the input features. The ReLU activation function is employed in MLP after each convolutional layer. Channel attention in CBAM is the same as Squeeze and Excitation (SE) attention [26] if only average pooling is used.

Second-order Attention Network: For single image super-resolution, in [27], the authors presented a second-order channel attention module, abbreviated as SOCA, to learn feature interdependencies via second-order feature statistics. A covariance matrix (Σ) is first computed and normalized using the features map from the previous network layers to obtain discriminative representations. The symmetric positive semi-definite covariance matrix is decomposed into $\Sigma = U\Lambda U^T$, where U is orthogonal, and Λ is the diagonal matrix with non-increasing eigenvalues. The power of eigenvalues $\Sigma = U\Lambda^\alpha U^T$ help in achieving the attention mechanism, that is, if $\alpha < 1$, then the eigenvalues larger than 1.0 will nonlinearly shrink while stretching others. The authors chose $\alpha < \frac{1}{2}$ based on previous work [28]. The subsequent attention mechanism is similar to SE [21] as shown in Figure 3(c), but instead of providing first-order statistics (*i.e.* global average pooling), the authors furnished second-order statistics (*i.e.* global covariance pooling).

High-Order Attention: To encode global information and contextual representations, [29], Ding *et al.* proposed High-order Attention (HA) with adaptive receptive fields and dynamic weights. HA mainly constructs a feature map for each pixel, including the relationships to other pixels. HA is required to address the issue of fixed-shape receptive fields that cause false prediction in case of high-shape objects *i.e.*, similar shape objects. Specifically, after calculating the attention maps for each pixel, graph transduction is used to form the final feature map. This feature representation is used to update each pixel position by using the weighted sum of contextual information. High-order attention maps are calculated using Hadamard product [30], [31]. It is classified as channel attention because it is generate attention scores from channels as in SE [26].

Harmonious attention: proposes a joint attention module of soft pixel and hard regional attentions [32]. The main idea is to tackle the limitation of the previous attention modules in person Re-Identification by learning attention selection and feature representation jointly and hence solving the issue of misalignment calibration caused by constrained attention mechanisms [33], [34], [35], [36]. Specifically, harmonious attention learns two types of soft attention (spatial and channel) in one branch and hard attention in the other one. Moreover, it proposes cross-interaction attention harmonizing between these two attention types as shown in Figure 3(i).

Auto Learning Attention: Ma *et al.* [57] introduced a novel idea for designing attention automatically. The module, named Higher-Order Group Attention (HOGA), is in the form of a Directed Acyclic Graph (DAG) [69], [70], [71], [72] where each node represents a group and each edge represents a heterogeneous attention operation. There is a sequential connection between the nodes to represent hybrids of attention operations. Thus, these connections can be represented as K-order attention modules, where K is the

number of attention operations. DARTS [73] is customized to facilitate the search process efficiently. This auto-learning module can be integrated into legacy architectures and performs better than manual ones. However, the core idea of attention modules remains the same as the previous architectures *i.e.* SE [26], CBAM [25], splat [23], mixed [74].

Double Attention Networks: Chen *et al.* [75] proposed Double Attention Network (A2-Nets), which attends over the input image in two steps. The first step gathers the required features using bilinear pooling to encode the second-order relationships between entities, and the second step distributes the features over the various locations adaptively. In this architecture, the second-order statistics, which are mostly lost with other functions such as average pooling of SE [26], of the pooled features are captured first by bilinear pooling. The attention scores are then calculated not from the whole image such as [3] but from a compact bag, hence, enriching the objects with the required context only. The first step *i.e.*, feature gathering uses the outer product $\sum_i a_i b_i^T$ then softmax is used for attending the discriminative features. The second step *i.e.*, distribution is based on complementing each location with the required features where their summation is 1. The complete design of A2-Nets is shown in Figure 3(d). Experimental comparisons demonstrated that A2-Net improves the performance better than SE and non-local networks, and is more efficient in terms of memory and time.

Dual Attention Network: Jun *et al.* [37] presented a dual attention network for scene segmentation composed of position attention and channel attention working in parallel. The position attention aims to encode the contextual features in local ones. The attention process is straightforward: the input features f_A are passed through three convolutional layers to generate three feature maps (f_B , f_C , and f_D), which are reshaped. Matrix multiplication is performed between the f_B and the transpose of f_C , followed by softmax to obtain the spatial attention map. Again, matrix multiplication is performed between the generated f_D features and the spatial attention map. Finally, the output is multiplied with a scalar and summed element-wise with the input features f_A as shown in Figure 3(f).

Although channel attention involves similar steps as position attention, it is different because the features are used directly without passing through convolutional layers. The input features f_A are reshaped, transposed, multiplied (*i.e.* $f_A \times f_A^T$), and then passed through the softmax layer to obtain the channel attention map. Moreover, the input features are multiplied with the channel attention map, followed by the element-wise summation, to give the final output as shown in Figure 3(e).

Frequency Channel Attention: Channel attention requires global average pooling as a pre-processing step. Qin *et al.* [76] argued that the global average pooling operation can be replaced with frequency components. The frequency attention views the discrete cosine transform as the weighted input sum with the cosine parts. As global average pooling is a particular case of frequency-domain feature decomposition, the authors use various frequency components of 2D discrete cosine transform, including the zero-frequency component, *i.e.* global average pooling.

TABLE 1

Summary of attention types along with their categorization, applications, strengths, and limitations. References to the original papers and links to Sections where they are discussed are also provided.

Type	Category	Section	References	Applications	Strengths	Limitations
Soft (Deterministic)	Channel	2.1.1	SE-Net [26], ECA-Net [22], CBAM [25], [32], A2-Net [29], Dual [37]	visual recognition, person re-identification, medical image segmentation, video recognition	<ul style="list-style-type: none"> • Easy to model • Differentiable gradients • Non-local operations • Encoding context • Improving the performance 	<ul style="list-style-type: none"> • Expensive in memory usage • High computation cost • Biased to softmax radial nature • Subject to attention collapse due to lack of diversity
	Spatial	2.1.2	CBAM [25], PFA [38], [39], [40]	Visual Recognition, domain adaptation, saliency detection		
	Self-attentions	2.1.3	Transformers [2], Image Transformers [41], [42], [43]	Visual Recognition, multi-modal tasks, video processing, low-level vision , video recognition , 3D analysis		
	Category-based	2.1.7	GAIN [44] [45]	explainable machine learning, person re-identification, semantic segmentation	<ul style="list-style-type: none"> • Providing gradient understanding • Does not require extra supervision 	<ul style="list-style-type: none"> • Extra computation • Used only for supervised classification
	Multi-modal	2.1.5	CAN [46], SCAN [47], Perceiver [48], Boosted [49]	few-shot classification, image-text matching, image captioning	<ul style="list-style-type: none"> • Benefiting visual-language-based applications • Providing attentive supervision signals • Achieving higher accuracy rates 	<ul style="list-style-type: none"> • Expensive in memory usage • High computation cost • Inherit the limitations of soft and hard attentions
	Arithmetic	2.1.4	Drop-out [50], Mirror [51], Reverse [52], Inverse [53], Reciprocal [54]	weakly-supervised object localization, line detection, semantic segmentation,	<ul style="list-style-type: none"> • Efficient methods • Simple ideas • Easy to implement • Enriching the semantics of the models 	<ul style="list-style-type: none"> • Limited to certain applications • Inability to scale up • Inherit the limitations of soft and hard attentions
Hard (Stochastic)	Logical	2.1.6	Recurrent [55], Sequential [56], [57], Permutation invariant [58]	image recognition, object detection and segmentation, adversarial image classification, image tagging, anomaly detection	<ul style="list-style-type: none"> • Overcoming the issues of soft attentions • Addressing hard attention disadvantages 	<ul style="list-style-type: none"> • Complex architectures • High computation cost • Iterative processing
	Statistical	2.2.1	Bayesian [59], Repulsive [60], Variational [61], [62]	visual-question answering, captioning, image translation, natural language processing		
	Reinforcement-based	2.2.2	RESA [63], [64], self-critic [65]	person re-identification, natural language processing	<ul style="list-style-type: none"> • Encoding context 	<ul style="list-style-type: none"> • Expensive in memory usage
	Gaussian	2.2.3	GatCluster [66], Uncertainty [67]	image clustering medical natural language processing	<ul style="list-style-type: none"> • Encoding higher-order interactions 	<ul style="list-style-type: none"> • High computation cost • Non-differentiable
	Clustering	2.2.4	Expectation Maximization [68], GatCluster [66]	semantic segmentation	<ul style="list-style-type: none"> • Diverse attention scores • Higher improvements 	<ul style="list-style-type: none"> • Gradient vanishing • Requires tricks for training

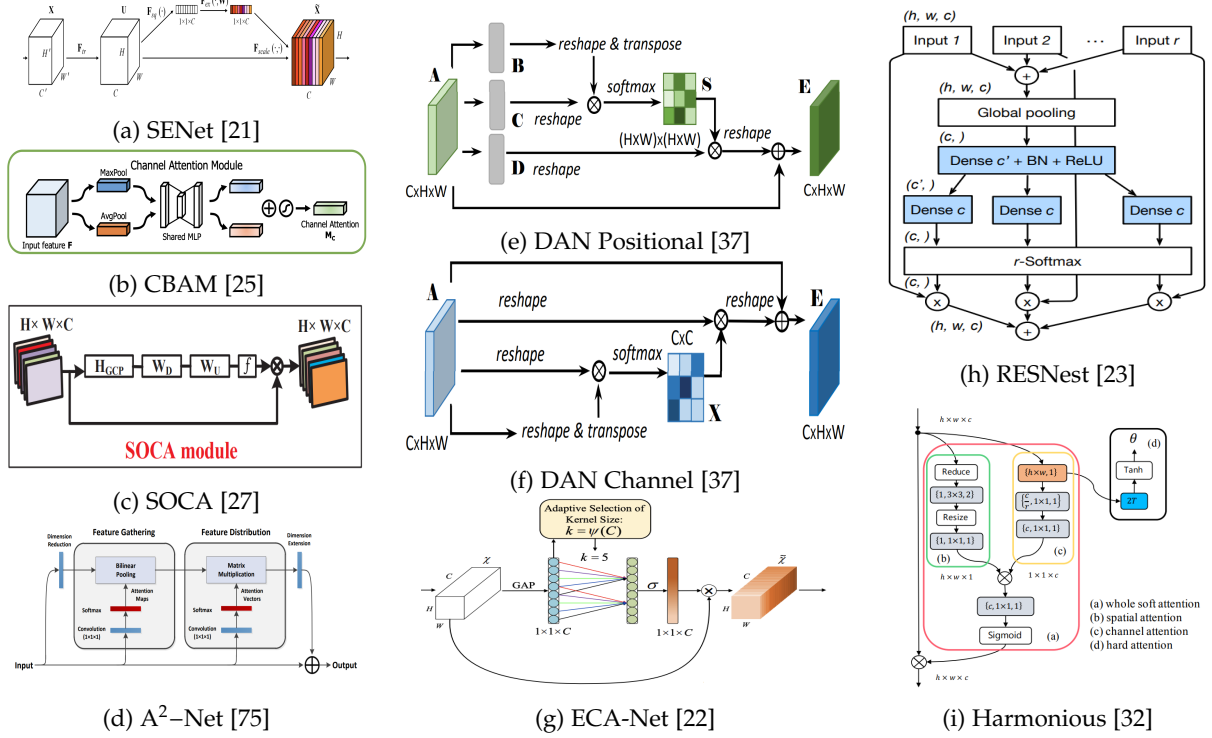


Fig. 3. Core structures of the channel-based attention methods. Different methods to generate the attention scores including squeeze and excitation [26], splitting and squeezing [23], calculating the second order [37] or efficient squeezing and excitation [22]. Images are taken from the original papers and are best viewed in color.

2.1.2 Spatial Attention

Different from channel attention that mainly generates channel-wise attention scores, spatial attention focuses on generating attention scores from spatial patches of the feature maps rather than the channels. However, the sequence of operations to generate the attentions are similar.

Spatial Attention in CBAM uses the inter-spatial feature relationships to complement the channel attention [25]. The spatial attention focuses on an informative part and is computed by applying average pooling and max pooling channel-wise, followed by concatenating both to obtain a single feature descriptor. Furthermore, a convolution layer on the concatenated feature descriptor is applied to generate a spatial attention map that encodes to emphasize or suppress. The feature map channel information is aggregated via average-pooled features and max-pooled features and then concatenated and convolved to generate a 2D spatial attention map. The overall process is shown in Figure 4(a) and computed as

$$f_{sp} = \sigma(\text{Conv}_{7 \times 7}([\text{MaxPool}(f); \text{AvgPool}(f)])), \quad (3)$$

where $\text{Conv}_{7 \times 7}$ denotes a convolution operation with the 7×7 kernel size and σ represents the sigmoid function.

Co-attention & Co-excitation: Hsieh *et al.* [77] proposed co-attention and co-excitation to detect all the instances that belong to the same target for one-shot detection. The main idea is to enrich the extracted feature representation using non-local networks that encode long-range dependencies and second-order interactions [3]. Co-excitation is based on squeeze-and-excite network [26] as shown in Figure 4(e).

While squeeze uses global average pooling [78] to reweight the spatial positions, co-excite serves as a bridge between the feature of query and target. Encoding high-contextual representations using co-attention and co-excitation show improvements in one-shot detector performance achieving state-of-the-art results.

Spatial Pyramid Attention Network abbreviated as SPAN [79], was proposed for localizing multiple types of image manipulations. It is composed of three main blocks *i.e.* feature extraction (head) module, pyramid spatial attention module, and decision (tail) module. The head module employs the Wider & Deeper VGG Network as the backbone, while Bayer and SRM layers extract features from visual artifacts and noise patterns. The spatial relationship of the pixels is achieved through five local self-attention blocks applied recursively, and to preserve the details, the input of the self-attention is added to the output of the block as shown in Figure 4(c). These features are then fed into the final tail module of 2D convolutional blocks to generate the output mask after employing a sigmoid activation.

Spatial-Spectral Self-Attention: Figure 4(d) shows the architecture of spatial-spectral self-attention which is composed of two attention modules, namely, spatial attention and spectral attention, both utilizing self-attention.

- 1) **Spatial Attention:** To model the non-local region information, Meng *et al.* [40] utilize a 3×3 kernel to fuse the input features indicating the region-based correlation followed by a convolutional network mapping the fused features into Q & K. The kernel number indicates the heads' number and the size denotes the dimension. Moreover, the dimension-specified features from Q & K

build the related attention maps, then modulating the corresponding dimension in a sequence to achieve the order-independent property. Finally, to finish the spatial correlation modeling, the features are forwarded to a deconvolution layer.

- 2) **Spectral Attention:** First, the spectral channel samples are convolved with one kernel and flattened into a single dimension, set as the feature vector for that channel. The input feature is converted to Q & K, building attention maps for the spectral axis. The adjacent channels have a higher correlation due to the image patterns on the exact location, denoted via a spectral smoothness on the attention maps. The similarity is indicated by normalized cosine distance as spectral embedding where each similar score is scaled and summed with the coefficients in the attention maps, which is then to modulate "Value" in self-attention, inducing spectral smoothness constraint.

Pixel-wise Contextual Attention: (PiCANet) [55] aims to learn accurate saliency detection. PiCANet generates a map at each pixel over the context region and constructs an accompanied contextual feature to enhance the feature representability at the local and global levels. To generate global attention, each pixel needs to "see" via ReNet [80] with four recurrent neural networks sweeping horizontally and vertically. The contexts from directions, using biLSTM, are blended propagating information of each pixel to all other pixels. Next, a convolutional layer transforms the feature maps to different channels, further normalized by a softmax function used to weight the feature maps. The local attention is performed on a local neighborhood forming a local feature cube where each pixel needs to "see" every other pixel in the local area using a few convolutional layers having the same receptive field as the patch. The features are then transformed to channel and normalized using softmax, which are further weighted summed to get the final attention.

Pyramid Feature Attention extracts features from different levels of VGG [38]. The low-level features extracted from lower layers of VGG are provided to the spatial attention mechanism [25], and the high-level features obtained from the higher layers are supplied to a channel attention mechanism [25]. The term feature pyramid attention originates from the VGG features obtained from different layers.

Spatial Attention Pyramid: For unsupervised domain adaptation, Li *et al.* [39] introduced a spatial attention pyramid that takes features from multiple average pooling layers with various sizes operating on feature maps. These features are forwarded to spatial attention followed by channel-wise attention. All the features after attention are concatenated to form a single semantic vector.

Region Attention Network: (RANet) [81] was proposed for semantic segmentation. It consists of novel network components, the Region Construction Block (RCB) and the Region Interaction Block (RIB), for constructing the contextual representations as illustrated in Figure 4(b). The RCB analyzes the boundary score and the semantic score maps jointly to compute the attention region score for each image pixel pair. High attention score indicates that the pixels are

from the same object region, dividing the image into various object regions. Subsequently, the RIB takes the region maps and selects the representative pixels in different regions where each representative pixel receives the context from other pixels to effectively represent the object region's local content. Furthermore, capturing the spatial and category relationship between various objects communicating the representative pixels in the different regions yields the global contextual representation to augment the pixels, eventually forming the contextual feature map for segmentation.

2.1.3 Self-attention

Self-attention, also known as *intra-attention*, is an attention mechanism that encodes the relationships between all the input entities. It is a process that enables input sequences to interact with each other and aggregate the attention scores, which illustrate how similar they are. The main idea is to replicate the feature maps into three copies and then measure the similarity between them. Apart from channel- and spatial-wise attention that use the physical feature maps, self-attention replicates feature copies to measure long-range dependencies. However, self-attention methods use channels to calculate attention scores. Cheng *et al.* extracted the correlations between the words of a single sentence using Long-Short-Term Memory (LSTM) [42]. An attention vector is produced from each hidden state during the recurrent iteration, which attends all the responses in a sequence for this position. In [82], a decomposable solution was proposed to divide the input into sub-problems, which improved the processing efficiency compared to [42]. The attention vector is calculated as an alignment factor to the content (bag-of-words). Although these methods introduced the idea of self-attention, they are very expensive in terms of resources and do not consider contextual information. Also, RNN models process input sequentially; hence, it is difficult to parallelize or process large-scale schema efficiently.

Transformers: Vaswani *et al.* [2] proposed a new method, called transformers, based on the self-attention concept without convolution or recurrent modules. As shown in Figure 5 (f), it is mainly composed of encoder-decoder layers, where the encoder comprises of self-attention module followed by positional feed-forward layer and the decoder is the same as the encoder except that it has an encoder-decoder attention layer in between. Positional encoding is represented by a sine wave that incorporates the passage of time as input before the linear layer. This positional encoding serves as a generalization term to help recognize unseen sequences and encodes relative positions rather than absolute representations. Algorithm 1 shows the detailed steps of calculating self-attention (multi-head attention) using transformers. Although transformers have achieved much progress in the text-based models, it lacks in encoding the context of the sentence because it calculates the word's attention for the left-side sequences. To address this issue, Bidirectional Encoder Representations from Transformers (BERT) learns the contextual information by encoding both sides of the sentence jointly [83].

Standalone self-attention: As stated above, convolutional features do not consider the global information due to their local-biased receptive fields. Instead of augmenting

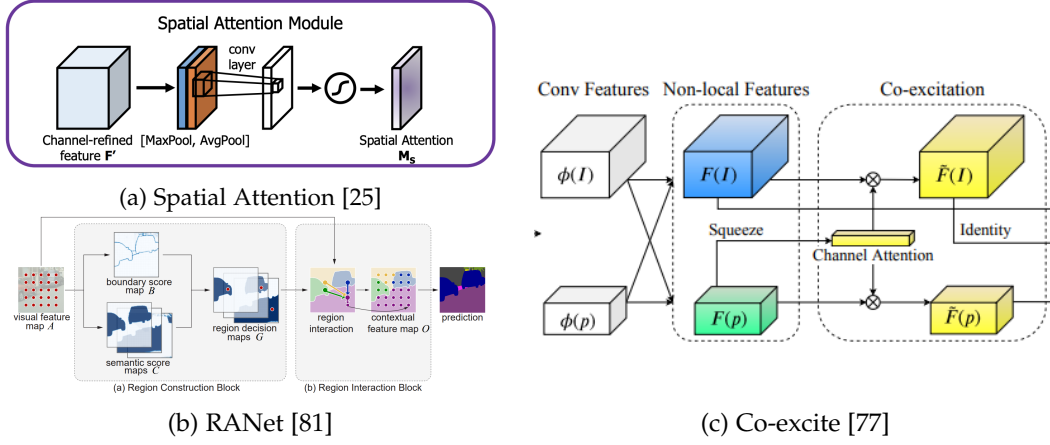


Fig. 4. The structures of the spatial-based attention methods, including RANet [81], and Co-excite [77]. These methods focus on attending to the most important parts in the spatial map. The images are taken from [25], [77], [81].

Algorithm 1: The main steps of generating self-attention by transformers (multi-head attention)

Input : set of sequences (x_1, x_2, \dots, x_n) of an entity $\mathbf{X} \in \mathbf{R}$

Output: attention scores of \mathbf{X} sequences.

- 1 Initialize weights: Key (\mathbf{W}_K), Query (\mathbf{W}_Q), Value (\mathbf{W}_V) for each input sequence.
- 2 Derive Key, Query, Value for each input sequence and its corresponding weight, such that $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_K$, $\mathbf{V} = \mathbf{X}\mathbf{W}_V$, respectively.
- 3 Compute attention scores by calculating the dot product between the query and key.
- 4 Compute the scaled-dot product attention for these scores and Values \mathbf{V} ,

$$\text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}.$$

- 5 repeat steps from 1 to 4 for all the heads

attentional features to the convolutional ones, Ramachandran *et al.* [43] proposed a fully-attentional network that replaces spatial convolutions with self-attentional modules. The convolutional stem (the first few convolutions) is used to capture spatial information. They designed a small kernel (e.g. $n \times n$) instead of processing the whole image simultaneously. This design built a computationally efficient model that enables processing images with their original sizes without downsampling. The computation complexity is reduced to $O(hwn^2)$, where h and w denotes height and width, respectively. A patch is extracted as a query along with each local patch, while the identity image is used as Value and Key. Calculating the attention maps follows the same steps as in Algorithm 1. Although stand-alone self-attention shows competitive results compared to convolutional models, it suffers from encoding positional information.

Clustered Attention: To address the computational inefficiency of transformers, Vyas *et al.* [84] proposed a clustered attention mechanism that relies on the idea that correlated queries follow the same distribution around Euclidean cen-

ters. Based on this idea, they use the K-means algorithm with fixed centers to group similar queries together. Instead of calculating the queries for attention, they are calculated for clusters' centers. Therefore, the total complexity is minimized to a linear form $O(qc)$, where q is the number of the queries while c is the cluster number.

Slot Attention: Locatello *et al.* [90] proposed slot attention, an attention mechanism that learns the objects' representations in a scene. In general, it learns to disentangle the image into a series of slots. As shown in Figure 5(d), each slot represents a single object. Slot attention module is applied to a learned representation $h \in \mathbb{R}^{W \times H \times D}$, where H is the height, W is the width, and D is the representation size. SAM has two main steps: learning n slots using an iterative attention mechanism and representing individual objects (slots). Inside each iteration, two operations are implemented: 1) slot competition using softmax followed by normalization according to slot dimension using this equation

$$a = \text{Softmax}\left(\frac{1}{\sqrt{D}}n(h).q(c)^T\right). \quad (4)$$

- 2) An aggregation process for the attended representations with a weighted mean

$$r = \text{Weightedmean}\left(a, v(h)\right), \quad (5)$$

where k, q, v are learnable variables as showed in [2]. Then, a feed-forward layer is used to predict the slot representations $s = f_c(r)$.

Slot attention is based on Transformer-like attention [2] on top of CNN-feature extractors. Given an image \mathbb{I} , the slot attention parses the scene into a set of slots, each one referring to an object (z, x, m) , where z is the object feature, x is the input image and m is the mask. In the decoders, convolutional networks are used to learn slot representations and object masks. The training process is guided by ℓ_2 -norm loss

$$\mathbb{L} = \left\| \left(\sum_{k=1}^K mx_k \right) - \mathbb{I} \right\|_2^2 \quad (6)$$

Following the slot-attention module, Li *et al.* developed an explainable classifier based on slot attentions [90]. This

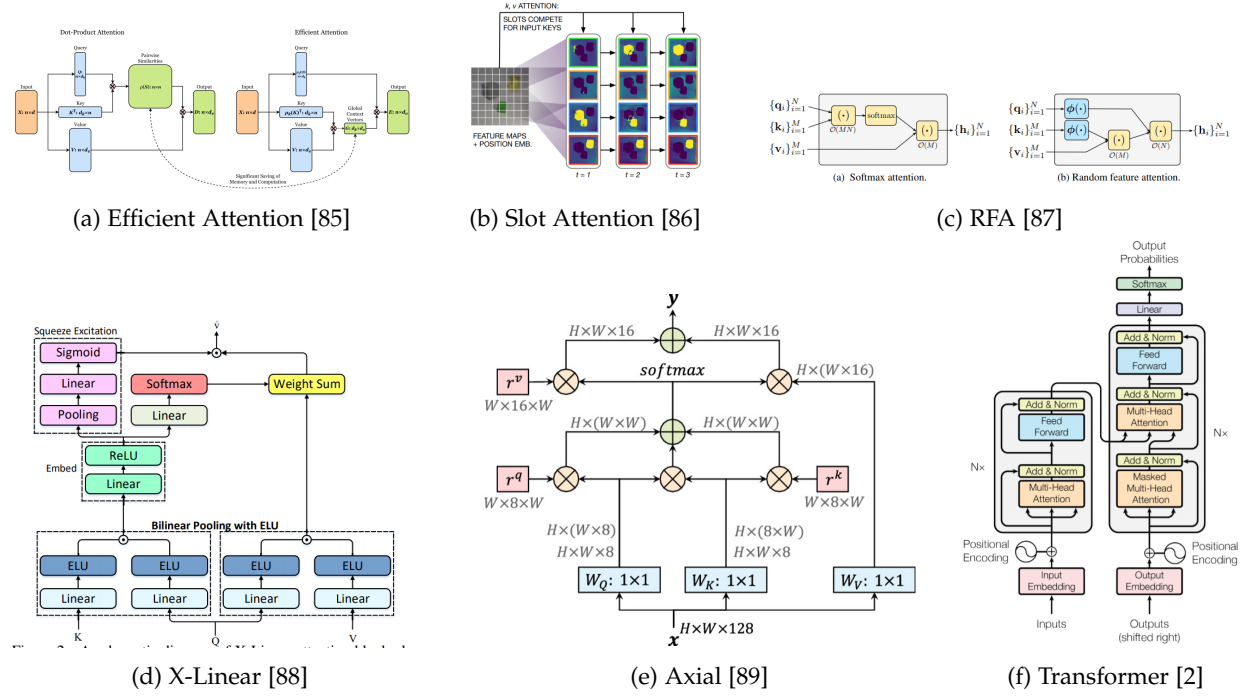


Fig. 5. The architectures of self-attention methods: Transformers [2], Axial attention [89], X-Linear [88], Slot [86] and RFA [87] (pictures taken from the corresponding articles). All of these methods are self-attention, which generate the scores from measuring the similarity between two maps of the same input. However, there is difference in the way of processing.

method aims to find the positive supports and negatives ones for a class l . In this way, a classifier can also be explained rather than being completely black-box. The primary entity of this work is xSlot, a variant of slot attention [86], which is related to a category and gives confidence for inclusion of this category in the input image.

Efficient Attention: Using Asymmetric Clustering (SMYRF) Daras *et al.* [91] proposed symmetric Locality Sensitive Hashing (LSH) clustering in a novel way to reduce the size of attention maps, therefore, developing efficient models. They observed that attention weights are sparse as well as the attention matrix is low-rank. As a result, pre-trained models pertain to decay in their values. In SMYRF, this issue is addressed by approximating attention maps through balanced clustering, produced by asymmetric transformations and an adaptive scheme. SMYRF is a drop-in replacement for pre-trained models for normal dense attention. Without retraining models after integrating this module, SMYRF showed significant effectiveness in memory, performance, and speed. Therefore, the feature maps can be scaled up to include contextual information. In some models, the memory usage is reduced by 50%. Although SMYRF enhanced memory usage in self-attention models, the improvement over efficient attention models is marginal (see Figure 5 (a)).

Random Feature Attention: Transformers have a major shortcoming with regards to time and memory complexity, which hinder attention scaling up and thus limiting higher-order interactions. Peng *et al.* [87] proposed reducing the space and time complexity of transformers from quadratic to linear. They simply enhance softmax approximation using random functions. Random Feature Attention (RFA) [92] uses a variant of softmax that is sampled from simple

distribution-based Fourier random features [93], [94]. Using the kernel trick $\exp(x.y) \approx \phi(x).\phi(y)$ of [95], softmax approximation is reduced to linear as shown in Figure 5 (c). Moreover, the similarity of RFA connections and recurrent networks help in developing a gating mechanism to learn recency bias [96], [97], [98]. RFA can be integrated into backbones easily to replace the normal softmax without much increase in the number of parameters, only 0.1% increase. Plugging RFA into a transformer show comparable results to softmax, while gating RFA outperformed it in language models. RFA executes 2× faster than a conventional transformer.

Non-local Networks: Recent breakthroughs in the field of artificial intelligence are mostly based on the success of Convolution Neural Networks (CNNs) [24], [99]. In particular, they can be processed in parallel mode and are inductive biases for the extracted features. However, CNNs fail to learn the context of the whole image due to their local-biased receptive fields. Therefore, long-range dependencies are disregarded in CNNs. In [3], Wang *et al.* proposed non-local networks to alleviate the bias of CNNs towards the local information and fuse global information into the network. It augments each pixel of the convolutional features with the contextual information, the weighted sum of the whole feature map. In this manner, the correlated patches in an image are encoded in a long-range fashion. Non-local networks showed significant improvement in long-range interaction tasks such as video classification [100] as well as low-level image processing [101], [102]. Non-local model attention in the network in a graphical fashion [103]. However, stacking multiple non-local modules in the same stage shows instability and ill-pose in the training

process [104]. In [105], Liu *et al.* uses non-local networks to form self-mutual attention between two modalities (RGB and Depth) to learn global contextual information. The idea is straightforward *i.e.* to sum the corresponding features before softmax normalization such that $\text{softmax}(f^r(\mathbb{X}^r) + \alpha^d \odot f^d(\mathbb{X}^d))$ for RGB attention and vice versa.

Non-Local Sparse Attention (NLSA): Mei *et al.* [106] proposed a sparse non-local network to combine the benefits of non-local modules to encode long-range dependencies and sparse representations for robustness. Deep features are split into different groups (buckets) with high inner correlations. Locality Sensitive Hashing (LSH) [107] is used to find similar features to each bucket. Then, the Non-Local block processes the pixel within its bucket and the similar ones. NLSA reduces the complexity to asymptotic linear from quadratic as well as uses the power of sparse representations to focus on informative regions only.

X-Linear Attention: Bilinear pooling is a calculation process that computes the outer product between two entities rather than the inner product [108], [109], [110], [111] and has shown the ability to encode higher-order interaction and thus encourage more discriminability in the models. Moreover, it yields compact models with the required details even though it compresses the representations [110]. In particular, bilinear applications has shown significant improvements in fine-grained visual recognition [112], [113], [114] and visual question answering [115]. As Figure 5(e) depicts, a low-rank bilinear pooling is performed between queries and keys, and hence, the 2^{nd} -order interactions between keys and queries are encoded. Through this query-key interaction, spatial-wise and channel-wise attention are aggregated with the values. The channel-wise attention is the same as squeeze-excitation attention [26]. The final output of the x-linear module is aggregated with the low-rank bilinear of keys and values [88]. They claimed that encoding higher interactions require only repeating the x-linear module accordingly (*e.g.* three iterative x-linear blocks for 4^{th} -order interactions). Modeling infinity-order interaction is also explained by the usage of h Exponential Linear Unit [116]. X-Linear attention module proposes a novel mechanism to attentions, different from transformer [2]. It is able to encode the relations between input tokens without positional encoding with only linear complexity as opposed to quadratic in transformer.

Axial-Attention: Wang *et al.* [89] proposed axial attention to encode global information and long-range context for the subject. Although conventional self-attention methods use fully-connected layers to encode non-local interactions, they are very expensive given their dense connections [2], [41], [117], [118]. Axial uses self-attention models in a non-local way without any constraints. Simply put, it factorizes the 2D self-attentions in two axes (width and height) of 1D-self attentions. This way, axial attention shows effectiveness in attending over wide regions. Moreover, unlike [119], [120], [121], axial attention uses positional information to include contextual information in an agnostic way. With axial attention, the computational complexity is reduced to $O(hwm)$. Also, axial attention showed competitive performance not only in comparison to full-attention models [43], [122], but convolutional ones as well [24], [123].

Efficient Attention Mechanism: Conventional attention mechanisms are built on double matrix multiplication that yields quadratic complexity $n \times n$ where n is the size of the matrix. Many methods propose efficient architectures for attention [85], [124], [125], [126]. In [85], Zhuoran *et al.* used the associative feature of matrix multiplication and suggested efficient attention. Formally, instead of using dot-product of the form $\rho(QK^T)V$, they process it in an efficient sequence $\rho_q(Q)(\rho_k(K)^T V)$ where ρ denotes a normalization step. Regarding the normalization of softmax, it is performed twice instead of once at the end. Hence, the complexity is reduced from quadratic $O(n^2)$ to linear $O(n)$. Through a simple change, the complexity of processing and memory usage are reduced to enable the integration of attention modules in large-scale tasks.

2.1.4 Arithmetic Attention

This part introduces arithmetic attention methods such as dropout, mirror, reverse, inverse, and reciprocal. We named it arithmetic because these methods are different from the above techniques even though they use their core. However, these methods mainly produce the final attention scores from simple arithmetic equations such as the reciprocal of the attention, *etc.*

Attention-based Dropout Layer: In weakly-supervised object localization, detecting the whole object without location annotation is a challenging task [127], [127], [128]. Choe *et al.* [129] proposed using the dropout layer to improve the localization accuracy through two steps: making the whole object location even by hiding the most discriminative part and attending over the whole area to improve the recognition performance. As Figure 6 (a) shows, ADL has two branches: 1) drop mask to conceal the discriminative part, which is performed by a threshold hyperparameter where values bigger than this threshold are set to zero and vice versa, and 2) importance map to give weight for the channels contributions by using a sigmoid function. Although the proposed idea is simple, experiments showed it is efficient (gained 15 % more than the state-of-the-art).

Mirror Attention: In a line detection application [51], Lee *et al.* developed mirrored attention to learn more semantic features. They flipped the feature map around the candidate line, and then concatenated the feature maps together. In case the line is not aligned, zero padding is applied.

Reverse Attention: Huang *et al.* [130] proposed the negative context (*e.g.* what is not related to the class) in training to learn semantic features. They were motivated by less discriminability between the classes in the high-level semantic representations and the weak response to the correct class from the latent representations. The network is composed of two branches, the first one learns discriminative features using convolutions for the target class, and the second one learns the reverse attention scores that are not associated with the target class. These scores are aggregated together to form the final attentions shown in Figure 6 (b). A deeper look inside the reverse attention shows that it is mainly dependent on negating the extracted features of convolutions followed by sigmoid $\text{sigmoid}(-F_{conv})$. However, for the purpose of convergence, this simple equation is

changed to be $\text{sigmoid}(\frac{1}{\text{ReLU}(F_{\text{conv}})+0.125} - 4)$. On semantic segmentation datasets, reverse attention achieved significant improvement over state-of-the-art. In a similar work, Chen *et al.* [52] proposed using reverse attention for salient object detection. The main intuition was to erase the final predictions of the network and hence learn the missing parts of the objects. However, the calculation method of attention scores is different from [130], whereas they used $1 - \text{sigmoid}(F_{i+1})$ where F_{i+1} denoted the features of the next stage.

2.1.5 Multi-modal attentions

As the name reveals, multi-modal attention is proposed to handle multi-modal tasks, using different modalities to generate attentions, such as text and image. It should be noted that some attention methods below, such as Perceiver [48] and Criss-Cross [131], [132], are transformer types [2], but are customized for multi-modal tasks by including text, audio, and image.

Cross Attention Network: In [46], a cross attention module (CAN) was proposed to enhance the overall discrimination of few-shot classification [133]. Inspired by the human behavior of recognizing novel images, the similarity between seen and unseen parts is identified first. CAN learns to encode the correlation between the query and the target object. As Figure 7 (a) shows, the features of query and target are extracted independently, and then a correlation layer computes the interaction between them using cosine distance. Next, 1D convolution is applied to fuse correlation (GAP is performed first) and attentions, followed by softmax normalization. The output is reshaped to give a single channel feature map to preserve spatial representations. Although, experiments show that CAN produces state-of-the-art results, it depends on non-learnable functions such as the cosine correlation. Also, the design is suitable for few-shot classification but is not general because it depends on two streams (query and target).

Criss-Cross Attention: The contextual information is still very important for scene understanding [131], [132]. Criss-cross attention proposed encoding the context of each pixel in the image in the criss-cross path. By building recurrent modules of criss-cross attention, the whole context is encoded for each pixel. This module is more efficient than non-local block [3] in memory and time, where the memory is reduced by 11x and GFLOPS reduced by 85%. Since this survey focuses on the core attention ideas, we show the criss-cross module in Figure 7 (b). Initially, three 1×1 convolutions are applied to the feature maps, whereas two of them are multiplied together (first map with each row of the second) to produce criss-cross attentions for each pixel. Then, softmax is applied to generate the attention scores, aggregated with the third convolution outcome. However, the encoded context captures only information in the criss-cross direction, and not the whole image. For this reason, the authors repeated the attention module by sharing the weights to form recurrent criss-cross, which includes the whole context. [134]

Perceiver Traditional: CNNs have achieved high performance in handling several tasks [24], [135], [136], however, they are designed and trained for a single domain rather

than multi-modal tasks [137], [138], [139]. Inspired by biological systems that understand the environment through various modalities simultaneously, Jaegle *et al.* proposed perceiver that leverages the relations between these modalities iteratively. The main concept behind perceiver is to form an attention bottleneck composed of a set of latent units. This solves the scale of quadratic processing, as in traditional transformer, and encourages the model to focus on important features through iterative processing. To compensate for the spatial context, Fourier transform is used to encode the features [140], [141], [142], [143]. As Figure 7 (c) shows, the perceiver is similar to RNN because of weights sharing. It composes two main components: cross attention to map the input image or input vector to a latent vector and transformer tower that maps the latent vector to a similar one with the same size. The architecture reveals that perceiver is an attention bottleneck that learns a mapping function from high-dimensional data to a low-dimensional one and then passes it to the transformer [2]. The cross-attention module has multi-byte attend layers to enrich the context, which might be limited from such mapping. This design reduces the quadratic processing $O(M^2)$ to $O(MN)$, where M is the sequence length and N is a hyperparameter that can be chosen smaller than M . Additionally, sharing the weights of the iterative attention reduces the parameters to one-tenth and enhances the model’s generalization.

Stacked Cross Attention: Lee *et al.* [47] proposed a method to attend between an image and a sentence context. Given an image and sentence, it learns the attention of words in a sentence for each region in the image and then scores the image regions by comparing each region to the sentence. This way of processing enables stacked cross attention to discover all possible alignments between text and image. Firstly, they compute image-text cross attention by a few steps as follows: a) compute cosine similarity for all image-text pairs [144] followed by ℓ_2 normalization [145], b) compute the weighted sum of these pairs attentions, where the image one is calculated by softmax [146], c) the final similarity between these pairs is computed using Log-SumExp pooling [147], [148]. The same steps are repeated to get the text-image cross attention, but the attention in the second step uses text-based softmax. Although stacked attention enriches the semantics of multi-modal tasks by attending text over image and vice versa, shared semantics might lead to misalignment in case of lack of similarity. With slight changes to the main concept, several works in various paradigms such as question answering and image captioning [149], [150], [151], [152], [153] used the stacked-cross attention.

Boosted Attention: While top-down attention mechanisms [154] fail to focus on regions of interest without prior knowledge, visual stimuli methods [155], [156] alone are not sufficient to generate captions for images. For this reason, in 7 (d), the authors proposed a boosted attention model to combine both of them in one approach to focus on top-down signals from the language and attend to the salient regions from stimuli independently. Firstly, they integrate stimuli attention with visual feature $I' = W I \circ \log(W_{\text{sal}} I + \epsilon)$, where I is the extracted features from the backbone, W_{sal} denotes the weight of the layer that produces stimuli attention, W

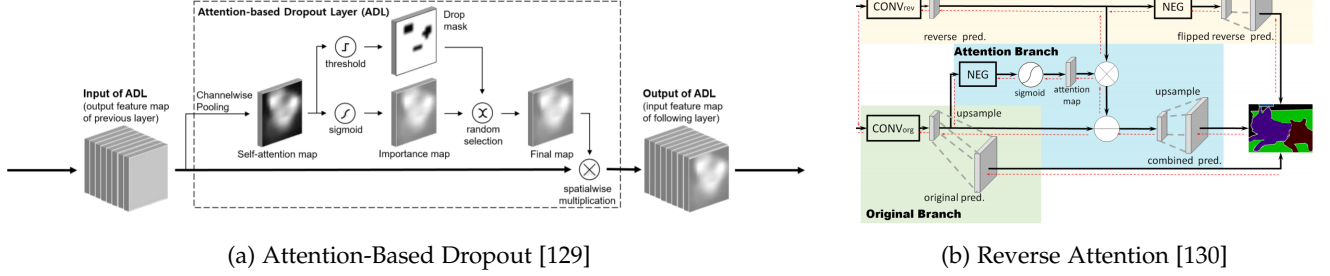


Fig. 6. The arithmetic-based attention methods i.e. Attention-based Dropout [129] and Reverse Attention [130]. Images are taken from the original papers. These methods use arithmetic operations to generate the attention scores such as reverse, dropout or reciprocal.

is the weight of the layer that output the visual feature layer. Boosted attention is achieved using the Hadamard product on I' . Their experiments work showed that boosted attention improved performance significantly.

2.1.6 Logical Attention

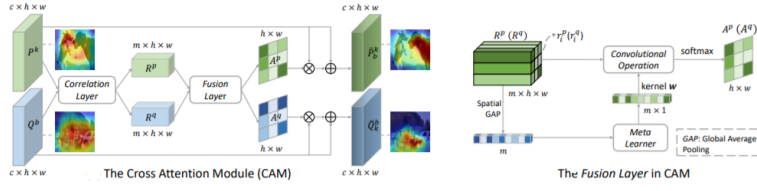
Similar to how human beings pay more attention to the crucial features, some methods have been proposed to use recurrences to encode better relationships. These methods rely on using RNNs or any type of sequential network to calculate the attentions. We named it logical methods because they use architectures similar to logic gates.

Sequential Attention Models: Inspired by the primate visual system, Zoran *et al.* [56] proposed soft, sequential, spatial top-down attention method (S3TA) to focus more on attended regions of an image [157] (as shown in Figure 8 (b)). At each step of the sequential process, the model queries the input and refines the total score based on spatial information in a top-down manner. Specifically, the backbone extracts features [24], [69], [123] channels that are split into keys and values. A Fourier transform encodes the spatial information for these two sets to preserve the spatial information from disappearing for later use. The main module is a top-down controller, which is a version of Long-Short Term Model (LSTM) [96], where its previous state is decoded as query vectors. The size of each query vector equals the sum of channels in keys and spatial basis. At each spatial location, the similarity between these vectors is calculated through the inner product, and then the softmax concludes the attention scores. These attention scores are multiplied by the values, and then the summation is taken to produce the corresponding answer vector for each query. All these steps are in the current step of LSTM and are then passed to the next step. Note that the input of the attention module is an output of the LSTM state to focus more on the relevant information as well as the attention map comprises only one channel to preserve the spatial information. Empirical evaluations show that attention is crucial for adversarial robustness because adversarial perturbations drag the object's attention away to degrade the model performance. Such an attention model proved its ability to resist strong attacks [158] and natural noises [159]. Although S3TA provides a novel method to empower the attention modules using recurrent networks, it is inefficient.

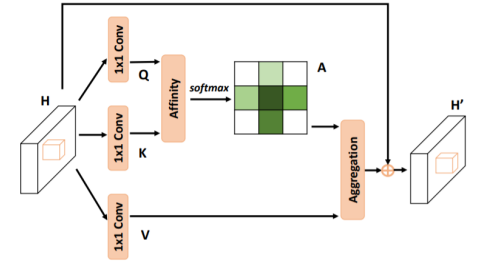
Permutation invariant Attention: Initially, Zaheer *et al.* [160] suggested handling deep networks in the form of sets rather than ordered lists of elements. For instance, performing pooling over sets of extracted features *e.g.* $\rho(\text{pool}(\phi(x_1), \phi(x_2), \dots, \phi(x_n)))$, where ρ and ϕ are continuous functions and pool can be the *sum* function. Formally, any set of deep learning features is invariant permutation if $f(\pi x) = \pi f(x)$. Hence, Lee *et al.* [58] proposed an attention-based method that processes sets of data. In [160], simple functions (*sum* or *mean*) are proposed to combine the different branches of the network, but they lose important information due to squashing the data. To address these issues, set transformer [58] parameterizes the pooling functions and provides richer representations that can encode higher-order interaction. They introduced three main contributions: a) Set Attention Block (SAB), which is similar to Multi-head Attention Block (MAB) layer [2], but without positional encoding and dropout; b) induced Set Attention Blocks (ISAB), which reduced complexity from $O(n^2)$ to $O(mn)$, where m is the size of induced point vectors and c) pooling by Multihead Attention (PMA) uses MAB over the learnable set of seed vectors.

Show, Attend and Tell: Xu *et al.* [62] introduced two types of attentions to attend to specific image regions for generating a sequence of captions aligned with the image using LSTM [161]. They used two types of attention: hard attention and soft attention. Hard attention is applied to the latent variable after assigning multinoulli distribution to learn the likelihood of $\log p(y|a)$, where a is the latent variable. By using multinoulli distribution and reducing the variance of the estimator, they trained their model by maximizing a variational lower bound as pointed out in [162], [163] provided that attentions sum to 1 at every point *i.e.* $\sum_i \alpha_{ti} = 1$, where α refers to attentions scores. For soft attention, they used softmax to generate the attention scores, but for $p(s_t|a)$ as in [50], where s_t is the extracted feature at this step. The training of soft attention is easily done by normal backpropagation and to minimize the negative log-likelihood of $-\log(p(y|a)) + \sum_i (1 - \sum_t \alpha_{ti})^2$. This model achieved the benchmark for visual captioning at the time as it paved the way for visual attention to progress.

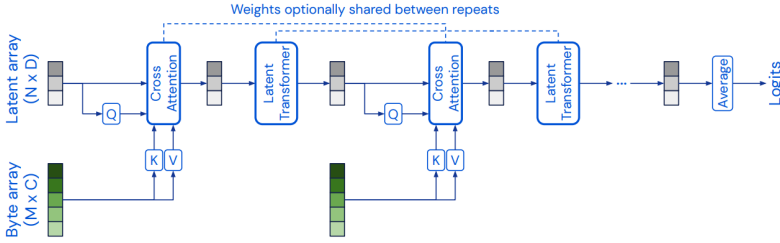
Kalman Filtering Attention: Liu *et al.* identified two main limitations that hinder using attention in various fields where there is insufficient learning or history [164]. These issues are 1) object's attention for input queries is covered



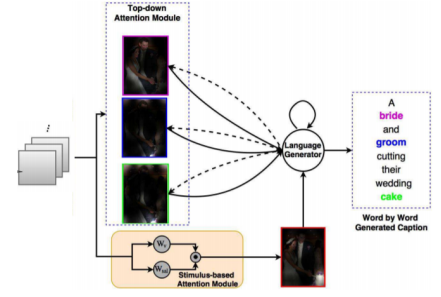
(a) Cross Attention [46]



(b) Criss-Cross Attention [134]



(c) Perceiver [48]



(d) Boosted Attention [49]

Fig. 7. Multi-modal attention methods consisting of Attention-based Perceiver [48], Criss-Cross [46], Boosted attention [49], Cross-attention module [46]. The mentioned methods employ multi-modalities to generate the attention scores. Images are taken from the original papers.

by past training; and 2) conventional attentions do not encode hierarchical relationships between similar queries. To address these issues, they proposed the use of Kalman filter attention. Moreover, KFAtt-freq to capture the homogeneity of the same queries, correcting the bias towards frequent queries.

Prophet Attention: In prophet attention [165], the authors noticed that the conventional attention models are biased and cause deviated focus to the tasks, especially in sequence models such as image captioning [166], [167] and visual grounding [168], [169]. Further, this deviation happens because attention models utilize previous input in a sequence to attend to the image rather than the outputs. As shown in Figure 8 (a), the model is attending on “yellow and umbrella” instead of “umbrella and wearing”. In a like-self-supervision way, they calculate the attention vectors based on the words generated in the future. Then, they guide the training process using these correct attentions, which can be considered a regularization of the whole model. Simply put, this method is based on summing the attentions of the post sequences in the same sentence to eliminate the impact of deviated focus towards the inputs. Overall, prophet attention addresses sequence-models biases towards history while disregarding the future.

2.1.7 Category-Based Attentions

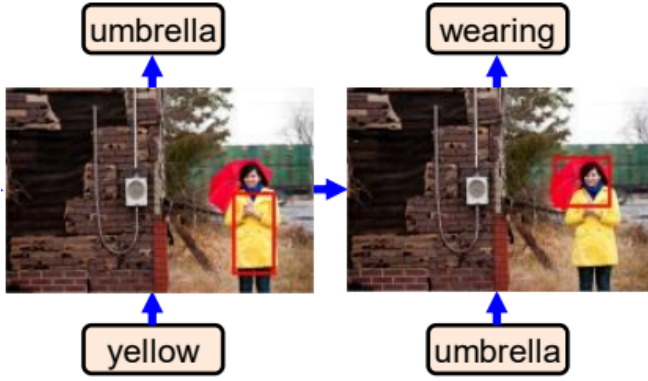
The above methods generate the attention scores from the features regardless of the presence of the class. On the other hand, some methods use class annotation to force the network to attend over specific regions.

Guided Attention Inference Network: In [44], the authors proposed class-aware attention, namely Guided Attention Inference Networks (GAIN), guided by the labels. Instead

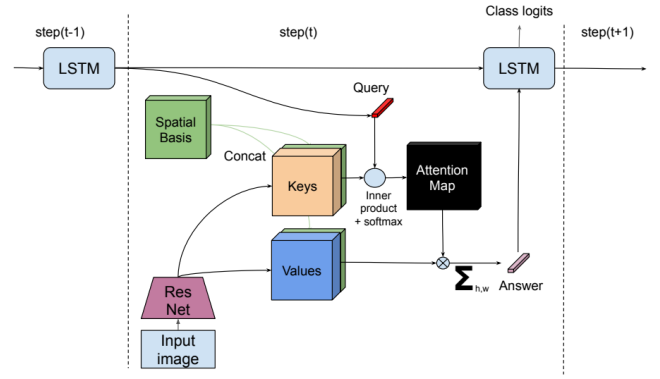
of focusing only on the most discriminative parts in the image [170], GAIN includes the contextual information in the feature maps. Following [171], GAIN obtains the attention maps from an inference branch, which are then used for training. As shown in Figure 9, through 2D-convolutions, global average pooling, and ReLU, the important features are extracted A^c for each class. Following this, the features of each class are obtained as $I - (T(A^c) \odot I)$ where \odot is matrix multiplication. $T(A^c) = \frac{1}{1 + \exp(-w(A^c - \sigma))}$ where σ is a threshold parameter and w is a scaling parameter. Their experiments showed that without recursive runs, GAINS gained significant improvement over the state-of-the-art.

Curriculum Enhanced Supervised Attention Network: The majority of attention methods are trained in a weakly supervised manner, and hence, the attention scores are still far from the best representations [2], [3]. In [45], the authors introduced a novel idea to generate a Supervised-Attention Network (SAN). Using the convolution layers, they defined the output of the last layer to be equal to the number of classes. Therefore, performing attention using global average pooling [78] yields a weight for each category. In a similar study, Fukui *et al.* proposed using a network composed of three branches to obtain class-specific attention scores: feature extractor to learn the discriminative features, attention branch to compute the attention scores based on a response model, perception to output the attention scores of each class by using the first two modules. The main objective was to increase the visual explanations [170] of the CNN networks as it showed significant improvements in various fields such as fine-grained recognition and image classification.

Attentional Class Feature Network: Zhang *et al.* [172] introduced ACFNet, a novel idea to exploit the contextual



(a) Prophet [165]



(b) S3TA [56]

Fig. 8. The core structure of logic-based attention methods such as Prophet attention [165] and S3TA [56] which are a type of attention that use logical networks such as RNN to infer the attention scores. Images are taken from the original papers and are best viewed in color.

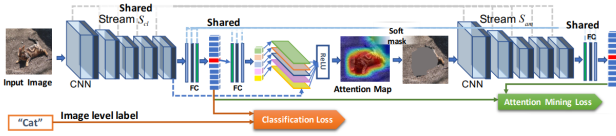


Fig. 9. An example of Guided Attention Inference Networks [44].

information for improving semantic segmentation. Unlike conventional methods that learn spatial-based global information [173], this contextual information is categorical-based, firstly presenting the class-center concept and then employing it to aggregate all the corresponding pixels to form a specific class representation. In the training phase, ground-truth labels are used to learn class centers, while coarse segmentation results are used in the test phase. Finally, class-attention maps are the results of class centers and coarse segmentation outcomes. The results show significant improvement for semantic segmentation using ACFNet.

2.2 Hard (Stochastic) Attention

Instead of using the weighted average of the hidden states, hard attention selects one of the states as the attention score. Proposing hard attention depends on answering two questions: (1) how to model the problem, and (2) how to train it without vanishing the gradients. In this part, hard attention methods are discussed as well as their training mechanisms. It includes a discussion of Bayesian attention, variational inference, reinforced, and Gaussian attentions. The main idea of Bayesian attention and variational one is to use latent random variables as attention scores. Reinforced attention replaces softmax with a Bernoulli-sigmoid unit [174], whereas Gaussian attention uses a 2D Gaussian kernel instead. Similarly, self-critic attention [65] employs a re-enforcement technique to generate the attention scores, whereas Expectation-Maximization uses EM to generate the scores.

2.2.1 Statistical-based attention

Bayesian Attention Modules (BAM) In contrast to the deterministic attention modules, Fan *et al.* [59] proposed a

stochastic attention method based on Bayesian-graph models. Firstly, keys and queries are aligned to form distributions parameters for attention weights, treated as latent random variables. They trained the whole model by reparameterization, which results from weight normalization by Lognormal or Weibull distributions. Kullback–Leibler (KL) divergence is used as a regularizer to introduce contextual prior distribution in the form of keys' functions. Their experiments illustrate that BAM significantly outperforms the state-of-the-art in various fields such as visual question answering, image captioning, and machine translation. However, this improvement happens on account of computational cost as well as memory usages. Compared to deterministic attention models, it is an efficient alternative in general, showing consistent effectiveness in language-vision tasks.

- **Bayesian Attention Belief Networks:** Zhang *et al.* [175] proposed using Bayesian Belief modules to generate attention scores given their ability to model high structured data along with uncertainty estimations. As shown in 10 (b), they introduced a simple structure to change any deterministic attention model into a stochastic one through four steps: 1) using Gamma distributions to build the decoder network 2) using Weibull distributions along with stochastic and deterministic paths for downward and upward, respectively 3) Parameterizing BABN distributions from the queries and keys of the current network 4) using evidence lower bound to optimize the encoder and decoder. The whole network is differentiable because of the existence of Weibull distributions in the encoder. In terms of accuracy and uncertainty, BABN proved improvement over the state-of-the-art in NLP tasks.
- **Repulsive Attention:** Multi-head attention [2] is the core of attention used in transformers. However, MHA may cause attention collapse when extracting the same features [60], [176], [177] and consequently, the discrimination power for feature representation will not be diverse. To address this issue, An *et al.* [60] adapted MHA to a Bayesian network with an underlying stochastic attention. MHA is considered a special case without

sharing parameters and using a particle-optimization sample to perform Bayesian inference on the attention parameter imposes attention repulsiveness [178]. Through this sampling method, each MHA is considered a sample seeking posterior distribution approximation, far from other heads.

Variational Attention In a study to improve the latent variable alignments, Deng *et al.* [61] proposed using variational attention mechanism. A latent variable is crucial because it encodes the dependencies between entities, whereas variational inference methods represent it in a stochastic manner [179], [180]. On the other hand, soft attention can encode alignments, but it has poor representation because of the nature of softmax. Using stochastic methods show better performance when optimized well [181], [182]. The main idea is to propose variational attention along with keeping the training tractable. They introduced two types of variational attention: categorical (hard) attention that uses amortized variational inference based on policy gradient and soft attention variance; relaxed (probabilistic soft attention) using Dirichlet distribution that allows attending over multiple sources. Regarding reparameterization, Dirichlet distribution is not parameterizable, and thus the gradient has high variances [183]. Inspired by [61], Bahuleyan *et al.* developed stochastic attention-based variational inference [184], but using a normal distribution instead of Dirichlet distribution. They observed that variational encoder-decoders should not have a direct connection; otherwise, traditional attentions serve as bypass connections.

2.2.2 Reinforcement-based Attention

Self-Critic Attention: Chen *et al.* [65] proposed a self-critic attention model that generates attention using an agent and re-evaluates the gain from this attention using the REINFORCE algorithm. They observed that most of the attention modules are trained in a weakly-supervised manner. Therefore, the attention maps are not always discriminative and lack supervisory signals during training [185]. To supervise the generation of attention maps, they used a reinforcement algorithm to guide the whole process. As shown in Figure 10 (a), the feature maps are evaluated to predict whether it needs self correctness or not.

Reinforced Self-Attention Network: Shen *et al.* [63] used a reinforced technique to combine soft and hard attention in one method. Soft attention has shown effectiveness in modeling local and global dependencies, which output from the dot-product similarity [6]. However, soft attention is based on the softmax function that assigns values to each item, even the non-attended ones, which weakens the whole attention. On the other hand, hard attention [12] attends to important regions or tokens only and disregards others. Despite its importance to textual tasks, it is inefficient in terms of time and differentiability [174]. Shen *et al.* [63] used hard attention to extract rich information and then feed it into soft attention for further processing. Simultaneously, soft attention is used to reward hard attention and hence stabilize the training process. Specifically, they used hard attention to encode tokens from the input in parallel while combining it with soft attention [186] without

any CNN/RNN modules (see Figure 10 (e)). In [64], reinforcement attention was proposed to extract better temporal context from video. Specifically, this attention module uses Bernoulli-sigmoid unit [174], a stochastic module. Thus, to train the whole system, REINFORCE algorithm is used to stabilize the gradients [183].

2.2.3 Gaussian-based Attention

Self Supervised Gaussian-Attention: Most soft-attention models use softmax to predict the attention of feature maps [2], [23], [41] which suffers from various drawbacks. In [187], Niu *et al.* proposed replacing the classical softmax with a Gaussian attention module. As shown in Figure 10 (d), they build a 2D Gaussian kernel to generate attention maps instead of softmax $K = e(-\frac{1}{\alpha}(u - \mu)^T \Sigma^{-1}(u - \mu))$ for each individual element, where $u = [x, y]^T$, $\mu = [\mu_x, \mu_y]^T$. A fully connected layer passes the extracted features, and then the Gaussian kernel is used to predict the attention scores. Using Gaussian kernels proved its effectiveness in discriminating the important features. Since it does not require any further learning steps, such as fully connected layers or convolutions, this significantly reduces the number of parameters. As stochastic training models need careful designs because of SGD mismatching [188], [189], [190], the Gaussian attention model developed binary classification loss that takes normalized logits to suppress the low scores and discriminate the high ones. This normalization uses a modified version of softmax, where the input is squared and divided by temperature value (e.g. batch size).

Uncertainty-Aware Attention: Since attention is generated without full supervision (*i.e.* in a weakly-supervised manner), it lacks full reliability [44]. To fix this issue, [67] proposed the use of uncertainty which is based on input. It generates varied attention maps according to the input and, therefore, learns higher variance for uncertain inputs. Gaussian distribution is used to handle attention weights, such that it gives small values in case of high confidence and vice versa [191]. Bayesian network is employed to build the model with variational inference as a solution [192], [193]. Note that this model is stochastic and SGD backpropagation flow can not work properly due to randomness [194]. For this reason, they used the reparameterization trick [195], [196] to train their model.

2.2.4 Clustering

Expectation Maximization attention: Traditional soft attention mechanisms can encode long-range dependencies by comparing each position to all positions, which is computationally very expensive [3]. In this regard, Li *et al.* [68] proposed using expectation maximization to build an attention method that iteratively forms a set of bases that compute the attention maps [197]. The main intuition is to use expectation maximization to select a compact basis set instead of using all the pixels as in [3], [39] (see Figure 10 (c)). These bases are regarded as the learning parameters, whereas the latent variables serve as the attention maps. The output is the weighted sum of bases, and the attention maps are the weights. The estimation step is defined by $z = \frac{\mathbb{K}(x_n, \mu_n)}{\sum_j \mathbb{K}(x_n, \mu_j)}$, where \mathbb{K} denotes a kernel function. The

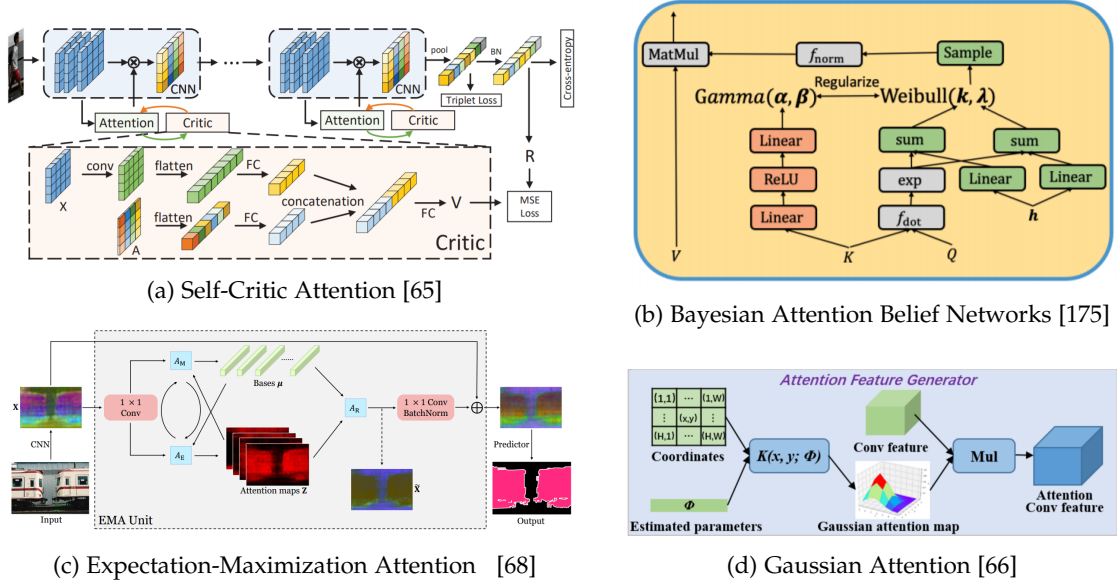


Fig. 10. Illustration of hard attention architectures. Building blocks of EMA [68], Gaussian [66], Self-critic [65] and Bayesian [175]. Images are taken from the original papers.

maximization step updates μ through data likelihood maximization such that $\mu = \frac{z_{nk}(x_n, \mu_n)}{\sum_j z_{jnk}}$. Finally, the features are multiplied by attention scores $\mathbb{X} = \mathbb{Z}\mu$. Since EMA is a stochastic model, training the whole model needs special care. Firstly, the authors average the μ over the mini-batch and update the maximization step to train it stably. Secondly, they normalize the Value of μ to be within $(1, T)$ by ℓ_2 -Norm. EMA has shown the ability to remove noisy representation and to give promising results after the third step. Also, it is worth noting that the complexity is reduced to a linear form $O(NK)$ from a quadratic one $O(N^2)$.

3 OPEN PROBLEMS AND CHALLENGES

Despite the performance improvement and interesting salient features of attention models, various challenges are associated with their practical settings in computer vision applications. The essential impediments include a requirement for high computational costs, significant amounts of training data, the efficiency of the model, and a cost-benefit analysis of performance improvement. In addition, there have also been some challenges to visualize and interpret attention blocks. This section provides an overview of these challenges and limitations, mentions some recent efforts to address those limitations, and highlights the open research questions.

Generalization: Attention models' generalization is a challenging task. Many of the proposed models are specific to the application underhand and only work well in the proposed settings. Whereas some models (e.g. channel and spatial attention) have performed better in classification since attention models are primarily designed for high-level tasks; they fail when applied directly on low-level vision tasks. Moreover, the data quality has a notable influence on the generalization and robustness of attention models. Thus, there is still a significant step to generalize pre-trained attention models on more generalized low-level vision tasks.

Efficiency: Efficiency of vision models is vital for many real-time computer vision applications. Unfortunately, current models focus more on performance than efficiency. Recently, self-attention has been successfully applied in transformers and shown to achieve better performance; however, at the cost of huge computational complexity e.g. the base ViT [9] has 18 billion FLOPs compared to the CNN models [198], [199] with 600 million FLOPs, achieving similar performance to process an image. Although fewer attempts such as Efficient Channel attention are made to make the attention models more efficient, they remain complex to train; hence, efficient models are required for deployment on real-time devices.

Multi-Model Data: Attention has been applied mainly on single domain data and in a single task setting. An important question is whether the attention model can fuse input data in a meaningful manner and exploit multiple label types (or tasks) in the data. Also, it is yet to be seen that attention models can leverage the various labels available, such as combining the point clouds and the RGB images of the KITTI dataset, to provide a meaningful performance. Similarly, attention models can also be used to know whether they can predict relationships between the labels, actions, and attributes in a unified manner.

Amount of Training Data: Attention models usually rely on much more training data to learn the important aspects, compared to simple non-attentional models. For example, self-attention employed in transformers needs to learn the invariance, translation *etc.* by themselves instead of non-attentional CNNs, where these properties are inbuilt due to operations such as pooling. The increase in data also means more training time and computational resources. Hence, an open question here is how to address this problem with more efficient attention models.

Performance Comparisons: Models that employ attentional blocks mostly compare their performance against the baseline without having the attention while ignoring other at-

tentional blocks. The lack of comparison between different attentional models provides little information about the actual performance improvement against other attentions. Therefore, there is a need to present a more in-depth analysis of the number of parameters increased versus the performance gain of different attentional models proposed in the literature.

4 CONCLUSIONS

In this paper, we reviewed more than 70 articles related to various attention mechanisms used in vision applications. We provided a comprehensive discussion of the attention techniques along with their strengths and limitations. We provided a restructuring of the existing attention mechanisms proposed in the literature into a hierarchical framework based on how they compute their attention scores. Choosing the attention score calculation to group the reviewed techniques has been effective in determining how the attention-based models are built and which training strategies are employed therein.

Although the capability of the developed attention-based techniques in modeling the salient features and boosting the performance is commendable, various challenges and open questions still remain unanswered, especially with the use of these techniques for computer vision tasks. We have listed these challenges and have highlighted research questions that still remain open. Despite some recent efforts introduced to cope with some of these limitations, we are still far from having solved the problems related to attention in vision. This survey will help researchers to better focus their efforts in addressing these challenges efficiently and in developing attention mechanisms that are better suited for vision-based applications.

5 ACKNOWLEDGMENTS

Professor Ajmal Mian is the recipient of an Australian Research Council Future Fellowship Award (project number FT210100268) funded by the Australian Government. We thank Professor Mubarak Shah for his useful comments that significantly improved the presentation of the survey.

REFERENCES

- [1] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International Conference on Machine Learning*. PMLR, 2017.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.
- [3] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018.
- [4] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019.
- [5] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *ICML*. PMLR, 2019.
- [6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv:1409.0473*, 2014.
- [7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014.
- [8] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.
- [9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [10] A. R. Kosiorek, A. Bewley, and I. Posner, "Hierarchical attentive recurrent tracking," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.
- [11] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, "Learn to pay attention," in *International Conference on Learning Representations*, 2018.
- [12] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015, pp. 2048–2057.
- [13] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *TNNLS*, 2019.
- [14] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *CSUR*, 2019.
- [15] S. Qiu, Y. Wu, S. Anwar, and C. Li, "Investigating attention mechanism in 3d point cloud object detection," in *International Conference on 3D Vision (3DV)*, 2021, pp. 403–412.
- [16] D. Hu, "An introductory survey on attention mechanisms in nlp problems," in *SAI Intelligent Systems Conference*. Springer, 2019.
- [17] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on visual transformer," *arXiv preprint arXiv:2012.12556*, 2020.
- [18] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *arXiv preprint arXiv:2101.01169*, 2021.
- [19] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, "Attention models in graphs: A survey," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2019.
- [20] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *NIPS*, vol. 28, 2015.
- [21] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.
- [22] W. Qilong, W. Banggu, Z. Pengfei, L. Peihua, Z. Wangmeng, and H. Qinghua, "Eca-net: Efficient channel attention for deep convolutional neural networks," in *CVPR*, 2020.
- [23] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, "Resnest: Split-attention networks," *arXiv:2004.08955*, 2020.
- [24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
- [25] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *ECCV*, 2018.
- [26] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.
- [27] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *CVPR*, June 2019.
- [28] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is second-order information helpful for large-scale visual recognition?" in *ICCV*, 2017.
- [29] F. Ding, G. Yang, J. Wu, D. Ding, J. Xu, G. Cheng, and X. Li, "High-order attention networks for medical image segmentation," in *MICCAI*, 2020.
- [30] R. A. Horn, "The hadamard product," in *Proc. Symp. Appl. Math*, vol. 40, 1990.
- [31] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang, "Hadamard product for low-rank bilinear pooling," *arXiv:1610.04325*, 2016.
- [32] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *CVPR*, June 2018.
- [33] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *ICPR*, 2014.
- [34] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *CVPR*, 2017, pp. 384–393.
- [35] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *ICCV*, 2015.
- [36] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *CVPR*, 2014.
- [37] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *CVPR*, 2019.

- [38] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *CVPR*, 2019.
- [39] C. Li, D. Du, L. Zhang, L. Wen, T. Luo, Y. Wu, and P. Zhu, "Spatial attention pyramid network for unsupervised domain adaptation," in *ECCV*, 2020.
- [40] Z. Meng, J. Ma, and X. Yuan, "End-to-end low cost compressive spectral imaging with spatial-spectral self-attention," in *ECCV*, 2020.
- [41] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *ICML*, 2018.
- [42] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *EMNLP*, 2016.
- [43] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *NeurIPS*, 2019.
- [44] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, "Tell me where to look: Guided attention inference network," in *CVPR*, 2018.
- [45] X. Zhu, J. Qian, H. Wang, and P. Liu, "Curriculum enhanced supervised attention network for person re-identification," *Signal Processing Letters*, vol. 27, 2020.
- [46] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, "Cross attention network for few-shot classification," *arXiv:1910.07677*, 2019.
- [47] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *ECCV*, 2018, pp. 201–216.
- [48] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," *arXiv:2103.03206*, 2021.
- [49] S. Chen and Q. Zhao, "Boosted attention: Leveraging human attention for image captioning," in *ECCV*, 2018, pp. 68–84.
- [50] P. Baldi and P. Sadowski, "The dropout learning algorithm," *Artificial intelligence*, vol. 210, pp. 78–122, 2014.
- [51] D. Jin, J. T. Lee, and C. S. Kim, "Semantic line detection using mirror attention and comparative ranking and matching," in *ECCV*, 2020.
- [52] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *ECCV*, 2018.
- [53] H. Zhang, H. Wang, Y. Cao, C. Shen, and Y. Li, "Robust watermarking using inverse gradient attention," *arXiv preprint arXiv:2011.10850*, 2020.
- [54] C. Xia, J. Li, J. Su, and Y. Tian, "Exploring reciprocal attention for salient object detection by cooperative learning," *arXiv preprint arXiv:1909.08269*, 2019.
- [55] N. Liu, J. Han, and M.-H. Yang, "Picanet: Learning pixel-wise contextual attention for saliency detection," in *IEEE CVPR*, 2018.
- [56] D. Zoran, M. Chrzanowski, P.-S. Huang, S. Goyal, A. Mott, and P. Kohli, "Towards robust image classification using sequential attention models," in *CVPR*, 2020.
- [57] B. Ma, J. Zhang, Y. Xia, and D. Tao, "Auto learning attention," in *NIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., 2020.
- [58] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *ICML*, 2019.
- [59] X. Fan, S. Zhang, B. Chen, and M. Zhou, "Bayesian attention modules," *arXiv:2010.10604*, 2020.
- [60] B. An, J. Lyu, Z. Wang, C. Li, C. Hu, F. Tan, R. Zhang, Y. Hu, and C. Chen, "Repulsive attention: Rethinking multi-head attention as bayesian inference," in *EMNLP*, 2020.
- [61] Y. Deng, Y. Kim, J. Chiu, D. Guo, and A. M. Rush, "Latent alignment and variational attention," in *NeurIPS*, 2018.
- [62] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015, pp. 2048–2057.
- [63] T. Shen, T. Zhou, G. Long, J. Jiang, S. Wang, and C. Zhang, "Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling," in *IJCAI*, 2018, pp. 4345–4352.
- [64] N. Karianakis, Z. Liu, Y. Chen, and S. Soatto, "Reinforced temporal attention and split-rate transfer for depth-based person re-identification," in *ECCV*, 2018, pp. 715–733.
- [65] G. Chen, C. Lin, L. Ren, J. Lu, and J. Zhou, "Self-critical attention learning for person re-identification," in *ICCV*, 2019.
- [66] C. Niu, J. Zhang, G. Wang, and J. Liang, "Gatcluster: Self-supervised gaussian-attention network for image clustering," in *ECCV*, 2020.
- [67] J. Heo, H. B. Lee, S. Kim, J. Lee, K. J. Kim, E. Yang, and S. J. Hwang, "Uncertainty-aware attention for reliable interpretation and prediction," in *NeurIPS*, 2018.
- [68] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *ICCV*, 2019.
- [69] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *ICML*, 2018.
- [70] S. Yang and D. Ramanan, "Multi-scale recognition with dacnn," in *ICCV*, 2015.
- [71] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.
- [72] L. Wang and H. Sahbi, "Directed acyclic graph kernels for action recognition," in *ICCV*, 2013.
- [73] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *ICLR*, 2018.
- [74] B. Chen, W. Deng, and J. Hu, "Mixed high-order attention network for person re-identification," in *ICCV*, 2019.
- [75] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "A 2-nets: double attention networks," in *NeurIPS*, 2018, pp. 350–359.
- [76] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: Frequency channel attention networks," *arXiv:2012.11879*, 2020.
- [77] T.-I. Hsieh, Y.-C. Lo, H.-T. Chen, and T.-L. Liu, "One-shot object detection with co-attention and co-excitation," in *NIPS*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., 2019.
- [78] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv:1312.4400*, 2013.
- [79] X. Hu, Z. Zhang, Z. Jiang, S. Chaudhuri, Z. Yang, and R. Nevatia, "Span: Spatial pyramid attention network for image manipulation localization," in *ECCV*, 2020.
- [80] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio, "Renet: A recurrent neural network based alternative to convolutional networks," *arXiv:1505.00393*, 2015.
- [81] D. Shen, Y. Ji, P. Li, Y. Wang, and D. Lin, "Ranet: Region attention network for semantic segmentation," *NIPS*, 2020.
- [82] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *EMNLP*, 2016.
- [83] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.
- [84] A. Vyas, A. Katharopoulos, and F. Fleuret, "Fast transformers with clustered attention," in *NeurIPS*, 2020.
- [85] Z. Shen, M. Zhang, H. Zhao, S. Yi, and H. Li, "Efficient attention: Attention with linear complexities," in *WACV*, 2021.
- [86] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-centric learning with slot attention," *arXiv:2006.15055*, 2020.
- [87] H. Peng, N. Pappas, D. Yogatama, R. Schwartz, N. Smith, and L. Kong, "Random feature attention," in *ICLR*, 2021.
- [88] Y. Pan, T. Yao, Y. Li, and T. Mei, "X-linear attention networks for image captioning," in *CVPR*, 2020.
- [89] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-deeplab: Stand-alone axial-attention for panoptic segmentation," in *ECCV*, 2020.
- [90] L. Li, B. Wang, M. Verma, Y. Nakashima, R. Kawasaki, and H. Nagahara, "Scouter: Slot attention-based classifier for explainable image recognition," *arXiv:2009.06138*, 2020.
- [91] G. Daras, N. Kitaev, A. Odena, and A. G. Dimakis, "Smyrf: Efficient attention using asymmetric clustering," *arXiv:2010.05315*, 2020.
- [92] A. S. Rawat, J. Chen, X. Y. Felix, A. T. Suresh, and S. Kumar, "Sampled softmax with random fourier features," in *NeurIPS*, 2019.
- [93] A. Rahimi, B. Recht *et al.*, "Random features for large-scale kernel machines," in *NIPS*, vol. 3, no. 4, 2007.
- [94] J. Yang, V. Sindhwani, H. Avron, and M. Mahoney, "Quasi-monte carlo feature maps for shift-invariant kernels," in *ICML*, 2014.
- [95] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, 2008.
- [96] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, 1997.

- [97] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.
- [98] J. Schmidhuber, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks," *Neural Computation*, 1992.
- [99] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, 2015.
- [100] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv:1705.06950*, 2017.
- [101] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *CVPR*, vol. 2, 2005.
- [102] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *CVPR*, 2019.
- [103] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018. [Online]. Available: <https://openreview.net/forum?id=rjXmpikCZ>
- [104] Y. Tao, Q. Sun, Q. Du, and W. Liu, "Nonlocal neural networks, nonlocal diffusion and nonlocal modeling," in *NeurIPS*, 2018.
- [105] N. Liu, N. Zhang, and J. Han, "Learning selective self-mutual attention for rgb-d saliency detection," in *CVPR*, 2020.
- [106] Y. Mei, Y. Fan, and Y. Zhou, "Image super-resolution with non-local sparse attention," in *CVPR*, 2021, pp. 3517–3526.
- [107] A. Gionis, P. Indyk, R. Motwani *et al.*, "Similarity search in high dimensions via hashing," in *Vldb*, vol. 99, no. 6, 1999, pp. 518–529.
- [108] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang, "Hadamard product for low-rank bilinear pooling," in *ICLR*, 2017.
- [109] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *ICCV*, 2015.
- [110] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *CVPR*, 2016.
- [111] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," in *EMNLP*, 2016.
- [112] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *ICCV*, 2015.
- [113] C. Yu, X. Zhao, Q. Zheng, P. Zhang, and X. You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *ECCV*, 2018.
- [114] S. Kong and C. Fowlkes, "Low-rank bilinear pooling for fine-grained classification," in *CVPR*, 2017.
- [115] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering," in *ICCV*, 2017.
- [116] J. T. Barron, "Continuously differentiable exponential linear units," *arXiv:1704.07483*, 2017.
- [117] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.
- [118] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.
- [119] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *ICCV*, 2019.
- [120] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *arXiv:1906.05909*, 2019.
- [121] H. Hu, Z. Zhang, Z. Xie, and S. Lin, "Local relation networks for image recognition," in *ICCV*, 2019.
- [122] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *ICCV*, 2019.
- [123] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.
- [124] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *ICLR*, 2019.
- [125] L. Wu, X. Liu, and Q. Liu, "Centroid transformers: Learning to abstract with attention," *arXiv preprint arXiv:2102.08606*, 2021.
- [126] Y. J. Kim and H. Hassan, "Fastformers: Highly efficient transformer models for natural language understanding," in *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, 2020.
- [127] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *ICCV*, 2011, pp. 1307–1314.
- [128] R. Gokberk Cinbis, J. Verbeek, and C. Schmid, "Multi-fold mil training for weakly supervised object localization," in *CVPR*, 2014, pp. 2409–2416.
- [129] J. Choe and H. Shim, "Attention-based dropout layer for weakly supervised object localization," in *CVPR*, 2019, pp. 2219–2228.
- [130] Q. Huang, C. Wu, C. Xia, Y. Wang, and C. J. Kuo, "Semantic segmentation with reverse attention," in *BMVC*, 2017.
- [131] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *ECCV*, 2018, pp. 603–619.
- [132] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *CVPR*, 2018, pp. 7151–7160.
- [133] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *CVPR*, 2018.
- [134] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *ICCV*, 2019, pp. 603–612.
- [135] X. Chen, X.-T. Yuan, Q. Chen, S. Yan, and T.-S. Chua, "Multi-label visual classification with label exclusive context," in *ICCV*, 2011.
- [136] M. Hassanin, I. Radwan, N. Moustafa, M. Tahtali, and N. Kumar, "Mitigating the impact of adversarial attacks in very deep networks," *Applied Soft Computing*, 2021.
- [137] Y. Luo, Y. Wen, D. Tao, J. Gui, and C. Xu, "Large margin multi-modal multi-task feature extraction for image classification," *TIP*, 2015.
- [138] W. Xu, W. Liu, X. Huang, J. Yang, and S. Qiu, "Multi-modal self-paced learning for image classification," *Neurocomputing*, vol. 309, 2018.
- [139] E. Alberts, G. Tetteh, S. Trebesch, M. Bieth, A. Valentinitich, B. Wiestler, C. Zimmer, and B. H. Menze, "Multi-modal image classification using low-dimensional texture features for genomic brain tumor recognition," in *Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics*, 2017.
- [140] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.
- [141] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. Siegelbaum, A. J. Hudspeth, and S. Mack, *Principles of neural science*. McGraw-hill New York, 2000.
- [142] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic programming and evolvable machines*, vol. 8, no. 2, 2007.
- [143] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *ICML*, 2018.
- [144] A. Karpathy, A. Joulin, and L. Fei-Fei, "Deep fragment embeddings for bidirectional image sentence mapping," *arXiv:1406.5679*, 2014.
- [145] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *International conference on Multimedia*, 2017, pp. 1041–1049.
- [146] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NeurIPS*, 2015, pp. 577–585.
- [147] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition," *Signal Processing Magazine*, pp. 14–36, 2008.
- [148] Y. Huang, Q. Wu, C. Song, and L. Wang, "Learning semantic concepts and order for image and sentence matching," in *CVPR*, 2018.
- [149] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *CVPR*, 2017.
- [150] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *CVPR*, 2017, pp. 5659–5667.
- [151] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *CVPR*, 2016, pp. 21–29.
- [152] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018, pp. 6077–6086.

- [153] D.-K. Nguyen and T. Okatani, "Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering," in *CVPR*, 2018.
- [154] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *CVPR*, 2017, pp. 375–383.
- [155] H. R. Tavakoli, R. Shetty, A. Borji, and J. Laaksonen, "Paying attention to descriptions generated by image captioning models," in *ICCV*, 2017, pp. 2487–2496.
- [156] Y. Sugano and A. Bulling, "Seeing with humans: Gaze-assisted neural image captioning," *arXiv:1608.05203*, 2016.
- [157] A. Mott, D. Zoran, M. Chrzanowski, D. Wierstra, and D. J. Rezende, "Towards interpretable reinforcement learning using attention augmented agents," *arXiv:1906.02500*, 2019.
- [158] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.
- [159] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," *arXiv:1907.07174*, 2019.
- [160] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola, "Deep sets," in *NeurIPS*, 2017.
- [161] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv:1409.2329*, 2014.
- [162] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *NeurIPS*, 2014, pp. 2204–2212.
- [163] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," in *ICLR (Poster)*, 2015.
- [164] H. Liu, J. LU, X. Zhao, S. Xu, H. Peng, Y. Liu, Z. Zhang, J. Li, J. Jin, Y. Bao, and W. Yan, "Kalman filtering attention for user behavior modeling in ctr prediction," in *NIPS*, 2020.
- [165] F. Liu, X. Ren, X. Wu, S. Ge, W. Fan, Y. Zou, and X. Sun, "Prophet attention: Predicting attention with future attention," *NIPS*, vol. 33, 2020.
- [166] J.-Y. Pan, H.-J. Yang, P. Duygulu, and C. Faloutsos, "Automatic image captioning," in *ICME*, 2004.
- [167] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Computing Surveys*, 2019.
- [168] C. Deng, Q. Wu, Q. Wu, F. Hu, F. Lyu, and M. Tan, "Visual grounding via accumulated attention," in *CVPR*, 2018.
- [169] G. A. Sigurdsson, J.-B. Alayrac, A. Nematzadeh, L. Smaira, M. Malinowski, J. Carreira, P. Blunsom, and A. Zisserman, "Visual grounding in video for unsupervised word translation," in *CVPR*, 2020.
- [170] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016.
- [171] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017.
- [172] F. Zhang, Y. Chen, Z. Li, Z. Hong, J. Liu, F. Ma, J. Han, and E. Ding, "Acnet: Attentional class feature network for semantic segmentation," in *ICCV*, 2019, pp. 6798–6807.
- [173] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
- [174] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, pp. 229–256, 1992.
- [175] S. Zhang, X. Fan, B. Chen, and M. Zhou, "Bayesian attention belief networks," *arXiv preprint arXiv:2106.05251*, 2021.
- [176] A. Prakash, J. Storer, D. Florencio, and C. Zhang, "Repr: Improved training of convolutional filters," in *CVPR*, 2019.
- [177] S. Han, J. Pool, S. Narang, H. Mao, E. Gong, S. Tang, E. Elsen, P. Vajda, M. Paluri, J. Tran *et al.*, "Dsd: Dense-sparse-dense training for deep neural networks," *arXiv:1607.04381*, 2016.
- [178] Q. Liu and D. Wang, "Stein variational gradient descent: a general purpose bayesian inference algorithm," in *30th NeurIPS*, 2016.
- [179] H. Salimbeni, V. Dutordoir, J. Hensman, and M. Deisenroth, "Deep gaussian processes with importance-weighted variational inference," in *ICML*, 2019.
- [180] I. Drori, "Deep variational inference," in *Handbook of Variational Methods for Nonlinear Geometric Data*, 2020.
- [181] J. W.-B. Lin and J. D. Neelin, "Toward stochastic deep convective parameterization in general circulation models," *Geophysical research letters*, vol. 30, no. 4, 2003.
- [182] H. Wang and D.-Y. Yeung, "A survey on bayesian deep learning," *ACM Computing Surveys*, vol. 53, no. 5, 2020.
- [183] M. Jankowiak and F. Obermeyer, "Pathwise derivatives beyond the reparameterization trick," in *ICML*, 2018.
- [184] H. Bahuleyan, L. Mou, O. Vechtomova, and P. Poupart, "Variational attention for sequence-to-sequence models," in *COLING*, 2018.
- [185] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial intelligence and statistics*, 2015.
- [186] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," in *AAAI Conference on Artificial Intelligence*, 2018.
- [187] C. Niu, J. Zhang, G. Wang, and J. Liang, "Gatcluster: Self-supervised gaussian-attention network for image clustering," in *ECCV*. Springer, 2020.
- [188] G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, and M. Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks," in *ICASSP*, 2014.
- [189] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *ECCV*, 2016.
- [190] G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, and M. Bacchiani, "Asynchronous stochastic optimization for sequence training of deep neural networks," in *ICASSP*, 2014.
- [191] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *NeurIPS*, 2017.
- [192] N. L. Zhang and D. Poole, "A simple approach to bayesian network computations," in *Canadian Conference on AI*, 1994.
- [193] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, 2017.
- [194] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv:1312.6114*, 2013.
- [195] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," in *NeurIPS*, 2017.
- [196] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *NeurIPS*, 2015.
- [197] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977.
- [198] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *CVPR*, 2020.
- [199] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *ICCV*, 2019.