

# ATTENTIONRNN: A Structured Spatial Attention Mechanism

Siddhesh Khandelwal<sup>1,2</sup> and Leonid Sigal<sup>1,2,3</sup>

<sup>1</sup>University of British Columbia

<sup>2</sup>Vector Institute for AI

<sup>3</sup>Canada CIFAR AI Chair

{skhandel, lsigal}@cs.ubc.ca

## Abstract

Visual attention mechanisms have proven to be integrally important constituent components of many modern deep neural architectures. They provide an efficient and effective way to utilize visual information selectively, which has shown to be especially valuable in multi-modal learning tasks. However, all prior attention frameworks lack the ability to explicitly model structural dependencies among attention variables, making it difficult to predict consistent attention masks. In this paper we develop a novel structured spatial attention mechanism which is end-to-end trainable and can be integrated with any feed-forward convolutional neural network. This proposed AttentionRNN layer explicitly enforces structure over the spatial attention variables by sequentially predicting attention values in the spatial mask in a bi-directional raster-scan and inverse raster-scan order. As a result, each attention value depends not only on local image or contextual information, but also on the previously predicted attention values. Our experiments show consistent quantitative and qualitative improvements on a variety of recognition tasks and datasets; including image categorization, question answering and image generation.

## 1. Introduction

In recent years, computer vision has made tremendous progress across many complex recognition tasks, including image classification [17, 44], image captioning [4, 14, 38, 40], image generation [30, 41, 43] and visual question answering (VQA) [2, 5, 13, 22, 27, 31, 37, 39]. Arguably, much of this success can be attributed to the use of visual attention mechanisms which, similar to the human perception, identify the important regions of an image. Attention mechanisms typically produce a spatial mask for the given image feature tensor. In an ideal scenario, the mask is expected to have higher activation values over the features corresponding to the regions of interest, and lower activation values everywhere else. For tasks that are multi-modal in nature, like VQA, a query (e.g., a question) can additionally be used as an input to generate the mask. In such cases, the

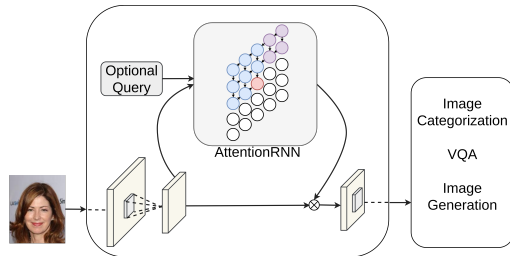


Figure 1: **AttentionRNN**. Illustration of the proposed structured attention network as a module for down stream task.

attention activation is usually a function of similarity between the corresponding encoding of the image region and the question in a pre-defined or learned embedding space.

Existing visual attention mechanisms can be broadly characterized into two categories: *global* or *local*; see Figure 2a and 2b respectively for illustration. Global mechanisms predict all the attention variables jointly, typically based on a dense representation of the image feature map. Such mechanisms are prone to overfitting and are only computationally feasible for low-resolution image features. Therefore, typically, these are only applied at the last convolutional layer of a CNN [22, 45]. The local mechanisms generate attention values for each spatial attention variable independently based on corresponding image region [5, 27, 28] (i.e., feature column) or with the help of local context [28, 35, 43]. As such, local attention mechanisms can be applied at arbitrary resolution and can be used at various places within a CNN network (e.g., in [28] authors use them before each sub-sampling layer and in [35] as part of each residual block). However, all the aforementioned models lack explicit structure in the generated attention masks. This is often exhibited by lack of coherence or sharp discontinuities in the generated attention activation values [28].

Consider a VQA model attending to regions required to answer the question, “Do the two spheres next to each other have the same color?”. Intuitively, attention mechanisms should focus on the two spheres in question. Furthermore, attention region corresponding to one sphere should inform the attention region estimates for the other, both in terms of shape and size. However, most traditional attention mechanisms have no ability to encode such dependencies. Recent

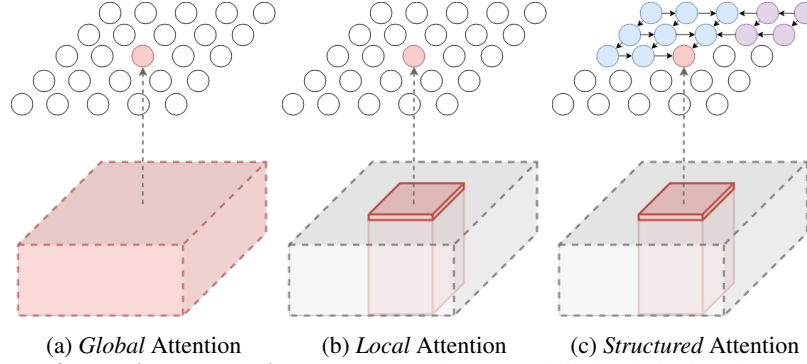


Figure 2: **Different types of attention mechanisms.** Compared are (a) *global* and (b) *local* attention mechanisms explored in prior works and proposed *structured* AttentionRNN architecture in (c).

modularized architectures [1, 10] are able to address some of these issues with attentive *reasoning*, but they are relevant only for a narrow class of VQA problems. Such models are inapplicable to scenarios involving self-attention [35] or generative architectures, where granular shape-coherent attention masks are typically needed [43].

In this paper, we argue that these challenges can be addressed by *structured* spatial attention. Such class of attention models can potentially encode arbitrary constraints between attention variables, be it top-down structured knowledge or local/global consistency and dependencies. To enforce this structure, we propose a novel attention mechanism which we refer to as AttentionRNN (see Figure 2c for illustration). We draw inspiration from the Diagonal BiLSTM architecture proposed in [33]. As such, AttentionRNN generates a spatial attention mask by traversing the image diagonally, starting from a corner at the top and going to the opposite corner at the bottom. When predicting the attention value for a particular image feature location, structure is enforced by taking into account: (i) local image context around the corresponding image feature location and, more importantly, (ii) information about previously generated attention values.

One of the key benefits of our model is that it can be used agnostically in any existing feed-forward neural network at one or multiple convolutional feature levels (see Figure 1). To support this claim, we evaluate our method on different tasks and with different backbone architectures. For VQA, we consider the Progressive Attention Network (PAN) [28] and Multimodal Compact Bilinear Pooling (MCB) [5]. For image generation, we consider the Modular Generative Adversarial Networks (MGAN) [43]. For image categorization, we consider the Convolutional Block Attention Module (CBAM) [35]. When we replace the existing attention mechanisms in these models with our proposed AttentionRNN, we observe higher overall performance along with better spatial attention masks.

**Contributions:** Our contributions can be summarized as follows: (1) We propose a novel spatial attention mechanism

which explicitly encodes structure over the spatial attention variables by sequentially predicting values. As a consequence, each attention value in a spatial mask depends not only on local image or contextual information, but also on previously predicted attention values. (2) We illustrate that this general attention mechanism can work with any existing model that relies on, or can benefit from, spatial attention; showing its effectiveness on a variety of different tasks and datasets. (3) Through experimental evaluation, we observe improved performance and better attention masks on VQA, image generation and image categorization tasks.

## 2. Related Work

### 2.1. Visual Attention

Visual attention mechanisms have been widely adopted in the computer vision community owing to their ability to focus on important regions in an image. Even though there is a large variety of methods that deploy visual attention, they can be categorized based on key properties of the underlying attention mechanisms. For ease of understanding, we segregate related research using these properties.

**Placement of attention in a network.** Visual attention mechanisms are typically applied on features extracted by a convolutional neural network (CNN). Visual attention can either be applied: (1) at the end of a CNN network, or (2) iteratively at different layers within a CNN network.

Applying visual attention at the end of a CNN network is the most straightforward way of incorporating visual attention in deep models. This has led to an improvement in model performance across a variety of computer vision tasks, including image captioning [4, 38, 40], image recognition [44], VQA [22, 37, 39, 45], and visual dialog [27].

On the other hand, there have been several approaches that iteratively apply visual attention, operating over multiple CNN feature layers [11, 28, 35]. Seo *et al.* [28] progressively apply attention after each pooling layer of a CNN network to accurately attend over target objects of various scales and shape. Woo *et al.* [35] use a similar approach, but instead apply two different types of attention - one that attends over feature channels and the other that attends over

the spatial domain.

**Context used to compute attention.** Attention mechanisms differ on how much information they use to compute the attention mask. They can be *global*, that is use all the available image context to jointly predict the values in an attention mask [22, 38]. As an example, [22] propose an attention mechanism for VQA where the attention mask is computed by projecting the image features into some latent space and then computing its similarity with the question.

Attention mechanisms can also be *local*, where-in attention for each variable is generated independently or using a corresponding local image region [5, 27, 28, 35, 43]. For example, [28, 35, 43] use a  $k \times k$  convolutional kernel to compute a particular attention value, allowing them to capture local information around the corresponding location.

None of the aforementioned works enforce structure over the generated attention masks. Structure over the values of an image, however, has been exploited in many autoregressive models trained to generate images. The next section briefly describes the relevant work in this area.

## 2.2. Autoregressive Models for Image Generation

Generative image modelling is a key problem in computer vision. In recent years, there has been significant work in this area [6, 15, 24, 26, 33, 42, 43]. Although most works use stochastic latent variable models like VAEs [15, 24] or GANs [6, 42, 43], autoregressive models [26, 32, 33] provide a more tractable approach to model the joint distribution over the pixels. These models leverage the inherent structure over the image, which enables them to express the joint distribution as a product of conditional distributions - where the value of the next pixel is dependent on all the previously generated pixels.

Van *et al.* [33] propose a PixelRNN network that uses LSTMs [9] to model this sequential dependency between pixels. They also introduce a variant, called PixelCNN, that uses CNNs instead of LSTMs to allow for faster computation. They later extend PixelCNN to allow for it to be conditioned on some query [32]. Finally, [26] propose further simplifications to the PixelCNN architecture to improve performance.

Our work draws inspiration from the PixelRNN architecture proposed in [33]. We extend PixelRNN to model structural dependencies within attention masks.

## 3. Approach

Given an input image feature  $\mathbf{X} \in \mathbb{R}^{h \times m \times n}$ , our goal is to predict a spatial attention mask  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , where  $h$  represents the number of channels, and  $m$  and  $n$  are the number of rows and the columns of  $\mathbf{X}$  respectively. Let  $\mathbf{X} = \{\mathbf{x}_{1,1}, \dots, \mathbf{x}_{m,n}\}$ , where  $\mathbf{x}_{i,j} \in \mathbb{R}^h$  be a column feature corresponding to the spatial location  $(i, j)$ . Similarly, let  $\mathbf{A} = \{a_{1,1}, \dots, a_{m,n}\}$ , where  $a_{i,j} \in \mathbb{R}$  be the attention

value corresponding to  $\mathbf{x}_{i,j}$ . Formally, we want to model the conditional distribution  $p(\mathbf{A} | \mathbf{X})$ . In certain problems, we may also want to condition on other auxiliary information in addition to  $\mathbf{X}$ , such as a question in VQA. While in this paper we formulate and model attention probabilistically, most traditional attention models directly predict the attention values, which can be regarded as a point estimate (or expected value) of  $\mathbf{A}$  under our formulation.

Global attention mechanisms [22, 45] predict  $\mathbf{A}$  directly from  $\mathbf{X}$  using a fully connected layer. Although this makes no assumptions on the factorization of  $p(\mathbf{A} | \mathbf{X})$ , it becomes intractable as the size of  $\mathbf{X}$  increases. This is mainly due to the large number of parameters in the fully connected layer.

Local attention mechanisms [27, 28, 35, 43], on the other hand, make strong independence assumptions on the interactions between the attention variables  $a_{i,j}$ . Particularly, they assume each attention variable  $a_{i,j}$  to be independent of other variables given some local spatial context  $\delta(\mathbf{x}_{i,j})$ . More formally, for local attention mechanisms,

$$p(\mathbf{A} | \mathbf{X}) \approx \prod_{i=1, j=1}^{i=m, j=n} p(a_{i,j} | \delta(\mathbf{x}_{i,j})) \quad (1)$$

Even though such a factorization improves tractability, the strong independence assumption often leads to attention masks that lack coherence and contain sharp discontinuities.

Contrary to local attention mechanisms, our proposed *AttentionRNN* tries to capture some of the structural dependencies between the attention variables  $a_{i,j}$ . We assume

$$p(\mathbf{A} | \mathbf{X}) = \prod_{i=1, j=1}^{i=m, j=n} p(a_{i,j} | \mathbf{a}_{<i,j}, \mathbf{X}) \quad (2)$$

$$\approx \prod_{i=1, j=1}^{i=m, j=n} p(a_{i,j} | \mathbf{a}_{<i,j}, \delta(\mathbf{x}_{i,j})) \quad (3)$$

where  $\mathbf{a}_{<i,j} = \{a_{1,1}, \dots, a_{i-1,j}\}$  (The blue and green region in Figure 3). That is, each attention variable  $a_{i,j}$  is now dependent on : (i) the local spatial context  $\delta(\mathbf{x}_{i,j})$ , and (ii) all the previous attention variables  $\mathbf{a}_{<i,j}$ . Note that Equation 2 is just a direct application of the chain rule. Similar to local attention mechanisms, and to reduce the computation overhead, we assume that a local spatial context  $\delta(\mathbf{x}_{i,j})$  is a sufficient proxy for the image features  $\mathbf{X}$  when computing  $a_{i,j}$ . Equation 3 describes the final factorization we assume.

One of the key challenges in estimating  $\mathbf{A}$  based on Equation 3 is to efficiently compute the term  $\mathbf{a}_{<i,j}$ . A straightforward solution is to use a recurrent neural network (e.g. LSTMs) to summarize the previously predicted attention values  $\mathbf{a}_{<i,j}$  into its hidden state. This is a common approach employed in many sequence prediction methods [3, 29, 34]. However, while sequences have a well defined ordering, image features can be traversed in multiple ways

due to their spatial nature. Naively parsing the image along its rows using an LSTM, though provides an estimate for  $\mathbf{a}_{<i,j}$ , fails to correctly encode the necessary information required to predict  $a_{i,j}$ . As an example, the LSTM will tend to forget information from the neighbouring variable  $a_{i-1,j}$  as it was processed  $n$  time steps ago.

To alleviate this issue, *AttentionRNN* instead parses the image in a diagonal fashion, starting from a corner at the top and going to the opposite corner in the bottom. It builds upon the Diagonal BiLSTM layer proposed by [33] to efficiently perform this traversal. The next section describes our proposed *AttentionRNN* mechanism in detail.

### 3.1. AttentionRNN

Our proposed structured attention mechanism builds upon the Diagonal BiLSTM layer proposed by [33]. We employ two LSTMs, one going from the top-left corner to the bottom-right corner ( $\mathcal{L}^l$ ) and the other from the top-right corner to the bottom-left corner ( $\mathcal{L}^r$ ).

As mentioned in Equation 3, for each  $a_{i,j}$ , our objective is to estimate  $p(a_{i,j} | \mathbf{a}_{<i,j}, \delta(\mathbf{x}_{i,j}))$ . We assume that this can be approximated via a combination of two distributions.

$$p(a_{i,j} | \mathbf{a}_{<i,j}) = \Gamma \langle p(a_{i,j} | \mathbf{a}_{<i,<j}), p(a_{i,j} | \mathbf{a}_{<i,>j}) \rangle \quad (4)$$

where  $\mathbf{a}_{<i,<j}$  is the set of attention variables to the top and left (blue region in Figure 3) of  $a_{i,j}$ ,  $\mathbf{a}_{<i,>j}$  is the set of attention variables to the top and right of  $a_{i,j}$  (green region in Figure 3), and  $\Gamma$  is some combination function. For brevity, we omit explicitly writing  $\delta(\mathbf{x}_{i,j})$ . Equation 4 is further simplified by assuming that all distributions are Gaussian.

$$\begin{aligned} p(a_{i,j} | \mathbf{a}_{<i,<j}) &\approx \mathcal{N}(\mu_{i,j}^l, \sigma_{i,j}^l) \\ p(a_{i,j} | \mathbf{a}_{<i,>j}) &\approx \mathcal{N}(\mu_{i,j}^r, \sigma_{i,j}^r) \\ p(a_{i,j} | \mathbf{a}_{<i,j}) &\approx \mathcal{N}(\mu_{i,j}, \sigma_{i,j}) \end{aligned} \quad (5)$$

where,

$$\begin{aligned} (\mu_{i,j}^l, \sigma_{i,j}^l) &= f_l(\mathbf{a}_{<i,<j}); (\mu_{i,j}^r, \sigma_{i,j}^r) = f_r(\mathbf{a}_{<i,>j}) \\ (\mu_{i,j}, \sigma_{i,j}) &= \Gamma(\mu_{i,j}^l, \sigma_{i,j}^l, \mu_{i,j}^r, \sigma_{i,j}^r) \end{aligned} \quad (6)$$

$f_l$  and  $f_r$  are fully connected layers. Our choice for the combination function  $\Gamma$  is explained in Section 3.2. For each  $a_{i,j}$ ,  $\mathcal{L}^l$  is trained to estimate  $(\mu_{i,j}^l, \sigma_{i,j}^l)$ , and  $\mathcal{L}^r$  is trained to estimate  $(\mu_{i,j}^r, \sigma_{i,j}^r)$ . We now explain the computation for  $\mathcal{L}^l$ .  $\mathcal{L}^r$  is analogous and has the same formulation.

$\mathcal{L}^l$  needs to correctly approximate  $\mathbf{a}_{<i,<j}$  in order to obtain a good estimate of  $(\mu_{i,j}^l, \sigma_{i,j}^l)$ . As we are parsing the image diagonally, from Figure 3 it can be seen that the following recursive relation holds,

$$\mathbf{a}_{<i,<j} = f(\mathbf{a}_{<i-1,<j}, \mathbf{a}_{<i,<j-1}) \quad (7)$$

That is, for each location  $(i, j)$ ,  $\mathcal{L}^l$  only needs to consider two attention variables- one above and the other to the left;

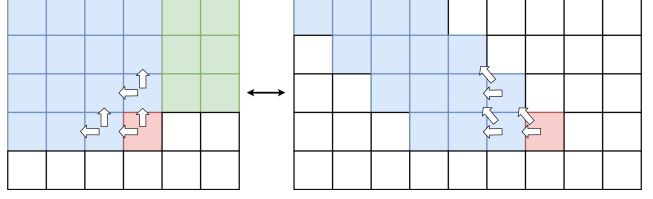


Figure 3: **Skewing operation.** This makes it easier to compute convolutions along the diagonal. The arrows indicate dependencies between attention values. To obtain the image on the right, each row of the left image is offset by one position with respect to its previous row.

[33] show that this is sufficient for it to be able to obtain information from all the previous attention variables.

To make computations along the diagonal easier, similar to [33], we first skew  $\mathbf{X}$  into a new image feature  $\hat{\mathbf{X}}$ . Figure 3 illustrates the skewing procedure. Each row of  $\mathbf{X}$  is offset by one position with respect to the previous row.  $\hat{\mathbf{X}}$  is now an image feature of size  $h \times m \times (2n - 1)$ . Traversing  $\mathbf{X}$  in a diagonal fashion from top left to bottom right is now equivalent to traversing  $\hat{\mathbf{X}}$  along its columns from left to right. As spatial locations  $(i - 1, j)$  and  $(i, j - 1)$  in  $\mathbf{X}$  are now in the same column in  $\hat{\mathbf{X}}$ , we can implement the recursion described in Equation 7 efficiently by performing computations on an entire column of  $\hat{\mathbf{X}}$  at once.

Let  $\hat{\mathbf{X}}_j$  denote the  $j^{th}$  column of  $\hat{\mathbf{X}}$ . Also, let  $\hat{\mathbf{h}}_{j-1}^l$  and  $\hat{\mathbf{c}}_{j-1}^l$  respectively denote the hidden and memory state of  $\mathcal{L}^l$  before processing  $\hat{\mathbf{X}}_j$ . Both  $\hat{\mathbf{h}}_{j-1}^l$  and  $\hat{\mathbf{c}}_{j-1}^l$  are tensors of size  $t \times m$ , where  $t$  is the number of latent features. The new hidden and memory state is computed as follows.

$$\begin{aligned} [\mathbf{o}_j, \mathbf{f}_j, \mathbf{i}_j, \mathbf{g}_j] &= \sigma \left( \mathbf{K}^h \circledast \hat{\mathbf{h}}_{j-1}^l + \mathbf{K}^x \circledast \hat{\mathbf{X}}_j^c \right) \\ \hat{\mathbf{c}}_j^l &= \mathbf{f}_j \odot \hat{\mathbf{c}}_{j-1}^l + \mathbf{i}_j \odot \mathbf{g}_j \\ \hat{\mathbf{h}}_j^l &= \mathbf{o}_j \odot \tanh(\mathbf{c}_j^l) \end{aligned} \quad (8)$$

Here  $\circledast$  represents the convolution operation and  $\odot$  represents element-wise multiplication.  $\mathbf{K}^h$  is a  $2 \times 1$  convolution kernel which effectively implements the recursive relation described in Equation 7, and  $\mathbf{K}^x$  is a  $1 \times 1$  convolution kernel. Both  $\mathbf{K}^h$  and  $\mathbf{K}^u$  produce a tensor of size  $4t \times m$ .  $\hat{\mathbf{X}}_j^c$  is the  $j^{th}$  column of the skewed local context  $\hat{\mathbf{X}}^c$ , which is obtained as follows.

$$\hat{\mathbf{X}}^c = \text{skew}(\mathbf{K}^c \circledast \mathbf{X}) \quad (9)$$

where  $\mathbf{K}^c$  is a convolutional kernel that captures a  $\delta$ -size context. Note that by defining  $\hat{\mathbf{X}}^c$  appropriately the proposed formulation, instead of using a simple convolution like in Equation 9, can employ more expressive networks such as U-Net [25] for  $\delta$ -context computation. Please refer to the supplementary material for additional details. For tasks that are multi-modal in nature, a query  $\mathbf{Q}$  can additionally be used to condition the generation of  $a_{i,j}$ . This allows



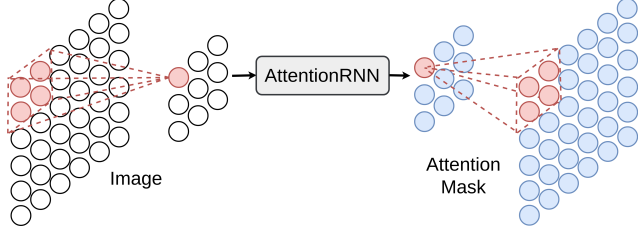


Figure 4: **Block AttentionRNN** for  $\gamma = 2$ . The input is first down-sized using a  $\gamma \times \gamma$  convolutional kernel. Attention is computed on this smaller map.

the model to generate different attention masks for the same image features depending on  $\mathbf{Q}$ . For example, in tasks like VQA, the relevant regions of an image will depend on the question asked. The nature of  $\mathbf{Q}$  will also dictate the encoding procedure. As an example, if  $\mathbf{Q}$  is a natural language question, it can be encoded using a LSTM layer.  $\mathbf{Q}$  can be easily incorporated into *AttentionRNN* by concatenating it with  $\hat{\mathbf{X}}^c$  before passing it to Equation 8.

Let  $\hat{\mathbf{h}}^l = \{\hat{\mathbf{h}}_1^l, \dots, \hat{\mathbf{h}}_{2n-1}^l\}$  be the set of all hidden states obtained from  $\mathcal{L}^l$ , and  $\mathbf{h}^l$  be the set obtained by applying the reverse skewing operation on  $\hat{\mathbf{h}}^l$ . For each  $a_{i,j}$ ,  $\mathbf{a}_{<i,<j}$  is then simply the  $(i, j)$  spatial element of  $\mathbf{h}^l$ .  $\mathbf{a}_{<i,>j}$  can be obtained by repeating the aforementioned process for  $\mathcal{L}^r$ , which traverses  $\mathbf{X}$  from top-right to bottom-left. Note that this is equivalent to running  $\mathcal{L}^r$  from top-left to bottom-right after mirroring  $\mathbf{X}$  along the column dimension, and then mirroring the output hidden states  $\mathbf{h}^r$  again. Similar to [33],  $\mathbf{h}^r$  is shifted down by one row to prevent  $\mathbf{a}_{<i,>j}$  from incorporating future attention values.

Once  $\mathbf{a}_{<i,<j}$  and  $\mathbf{a}_{<i,>j}$  are computed (as discussed above), we can obtain the Gaussian distribution for the attention variable  $\mathcal{N}(\mu_{i,j}, \sigma_{i,j})$  by following Equation 6. The attention  $a_{i,j}$  could then be obtained by either sampling a value from  $\mathcal{N}(\mu_{i,j}, \sigma_{i,j})$  or simply by taking the expectation and setting  $a_{i,j} = \mu_{i,j}$ . For most problems, as we will see in the experiment section, taking the expectation is going to be most efficient and effective. However, sampling maybe useful in cases where attention is inherently multimodal. Focusing on different modes using coherent masks might be more beneficial in such situations.

### 3.2. Combination Function

The choice of the combination function  $\Gamma$  implicitly imposes some constraints on the interaction between the distributions  $\mathcal{N}(\mu_{i,j}^l, \sigma_{i,j}^l)$  and  $\mathcal{N}(\mu_{i,j}^r, \sigma_{i,j}^r)$ . For example, assumption of independence would dictate a simple product for  $\Gamma$ , with resulting operations to produce  $(\mu_{i,j}, \sigma_{i,j})$  being expressed in closed form. However, it is clear that independence is unlikely to hold due to image correlations. To allow for a more flexible interaction between variables and combination function, we instead use a fully connected

layer to learn the appropriate  $\Gamma$  for a particular task.

$$\mu_{i,j}, \sigma_{i,j} = f_{comb}(\mu_{i,j}^l, \sigma_{i,j}^l, \mu_{i,j}^r, \sigma_{i,j}^r) \quad (10)$$

### 3.3. Block AttentionRNN

Due to the poor performance of LSTMs over large sequences, the AttentionRNN layer doesn't scale well to large image feature maps. We introduce a simple modification to the method described in Section 3.1 to alleviate this problem, which we refer to as Block AttentionRNN (BRNN).

BRNN reduces the size of the input feature map  $\mathbf{X}$  before computing the attention mask. This is done by splitting  $\mathbf{X}$  into smaller blocks, each of size  $\gamma \times \gamma$ . This is equivalent to down-sampling the original image  $\mathbf{X}$  to  $\mathbf{X}^{ds}$  as follows.

$$\mathbf{X}^{ds} = \mathbf{K}^{ds} \circledast \mathbf{X} \quad (11)$$

where  $\mathbf{K}^{ds}$  is a convolution kernel of size  $\gamma \times \gamma$  applied with stride  $\gamma$ . In essence, each value in  $\mathbf{X}^{ds}$  now corresponds to a  $\gamma \times \gamma$  region in  $\mathbf{X}$ .

Instead of predicting a different attention probability for each individual spatial location  $(i, j)$  in  $\mathbf{X}$ , BRNN predicts a single probability for each  $\gamma \times \gamma$  region. This is done by first computing the attention mask  $\mathbf{A}^{ds}$  for the down-sampled image  $\mathbf{X}^{ds}$  using AttentionRNN (Section 3.1), and then  $\mathbf{A}^{ds}$  is scaled up using a transposed convolutional layer to obtain the attention mask  $\mathbf{A}$  for the original image feature  $\mathbf{X}$ . Figure 4 illustrates the BRNN procedure.

BRNN essentially computes a coarse attention mask for  $\mathbf{X}$ . Intuitively, this coarse attention can be used in the first few layers of a deep CNN network to identify the key region blocks in the image. The later layers can use this coarse information to generate a more granular attention mask.

## 4. Experiments

To show the efficacy and generality of our approach, we conduct experiments over four different tasks: visual attribute prediction, image classification, visual question answering (VQA) and image generation. We highlight that our goal is not to necessarily obtain the absolute highest raw performance (although we do in many of our experiments), but to show improvements from integrating AttentionRNN into existing state-of-the-art models across a variety of tasks and architectures. Due to space limitations, all model architectures and additional visualizations are described in the supplementary material.

### 4.1. Visual Attribute Prediction

**Datasets.** We experiment on the synthetic MREF, MDIST and MBG datasets proposed in [28]. Figure 5 shows example images from the datasets. The images in the datasets are created from MNIST [18] by sampling five to nine distinct digits with different colors (green, yellow, white, red, or blue) and varying scales (between 0.5 and 3.0). The datasets have images of size 100 x 100 and only differ in how the

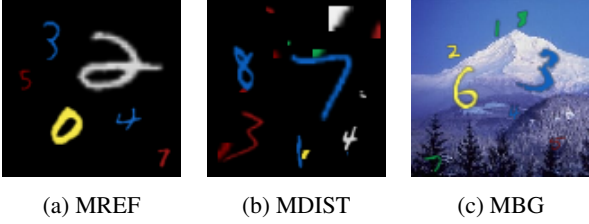


Figure 5: **Synthetic Dataset Samples.** Example images taken from the three synthetic datasets proposed in [27].

Attention	MREF	MDIST	MBG	Rel. Runtime
SAN [38]	83.42	80.06	58.07	1x
-CTX [28]	95.69	89.92	69.33	1.08x
CTX [28]	98.00	95.37	79.00	1.10x
ARNN <sub>ind</sub> <sup>~</sup>	98.72	96.70	83.68	4.73x
ARNN <sub>ind</sub>	98.58	96.29	84.27	
ARNN <sup>~</sup>	98.65	96.82	83.74	
ARNN	<b>98.93</b>	<b>96.91</b>	<b>85.84</b>	

Table 1: **Color prediction accuracy.** Results are in % on MREF, MDIST and MBG datasets. Our AttentionRNN-based model, CNN+ARNN, outperforms all the baselines.

background is generated. MREF has a black background, MDIST has a black background with some Gaussian noise, and MBG has real images sampled from the SUN Database [36] as background. The training, validation and test sets contain 30,000, 10,000 and 10,000 images respectively.

**Experimental Setup.** The performance of AttentionRNN (ARNN) is compared against two *local* attention mechanisms proposed in [28], which are referred as -CTX and CTX. ARNN assumes  $a_{i,j} = \mu_{i,j}, \delta = 3$ , where  $\mu_{i,j}$  is defined in Equation 10. To compute the attention for a particular spatial location  $(i, j)$ , CTX uses a  $\delta = 3$  local context around  $(i, j)$ , whereas -CTX only uses the information from the location  $(i, j)$ . We additionally define three variants of ARNN: i) ARNN<sup>~</sup> where each  $a_{i,j}$  is sampled from  $\mathcal{N}(\mu_{i,j}, \sigma_{i,j})$ , ii) ARNN<sub>ind</sub> where the combination function  $\Gamma$  assumes the input distributions are independent, and iii) ARNN<sub>ind</sub><sup>~</sup> where  $\Gamma$  assumes independence and  $a_{i,j}$  is sampled. The soft attention mechanism (SAN) proposed by [38] is used as an additional baseline. The same base CNN architecture is used for all the attention mechanisms for fair comparison. The CNN is composed of four stacks of  $3 \times 3$  convolutions with 32 channels followed by  $2 \times 2$  max pooling layer. SAN computes attention only on the output of the last convolution layer, while -CTX, CTX and all variants of ARNN are applied after each pooling layer. Given an image, the models are trained to predict the color of the number specified by a query. Chance performance is 20%.

**Results.** Table 1 shows the color prediction accuracy of various models on MREF, MDIST and MBG datasets. It can be seen that ARNN and all its variants clearly outperform the other baseline methods. The difference in performance

Attention	Corr.	Scale				
		0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0
SAN [38]	0.15	53.05	74.85	72.18	59.52	54.91
-CTX [28]	0.28	68.20	76.37	73.30	61.60	57.28
CTX [28]	0.31	77.39	87.13	84.96	75.59	63.72
ARNN <sub>ind</sub> <sup>~</sup>	0.36	82.23	89.41	86.46	84.52	81.35
ARNN <sub>ind</sub>	0.34	82.89	89.47	88.34	84.22	80.00
ARNN <sup>~</sup>	0.39	82.23	89.41	86.46	84.52	81.35
ARNN	<b>0.42</b>	<b>84.45</b>	<b>91.40</b>	<b>86.84</b>	<b>88.39</b>	<b>82.37</b>

Table 2: **Mask Correctness and Scale experiment on MBG.** The “Corr.” column lists the mask correctness metric proposed by [20]. The “Scale” column shows the color prediction accuracy in % for different scales.

is amplified for the more noisy MBG dataset, where ARNN is 6.8% better than the closest baseline. ARNN<sub>ind</sub> performs poorly compared to ARNN, which furthers the reasoning of using a neural network to model  $\Gamma$  instead of assuming independence. Similar to [28], we also evaluate the models on their sensitivity to the size of the target. The test set is divided into five uniform scale intervals for which model accuracy is computed. Table 2 shows the results on the MBG dataset. ARNN is robust to scale variations and performs consistently well on small and large targets. We also test the correctness of the mask generated using the metric proposed by [20], which computes the percentage of attention values in the region of interest. For models that apply attention after each pooling layer, the masks from different layers are combined by upsampling and taking a product over corresponding pixel values. The results are shown for the MBG dataset in Table 2. ARNN is able to more accurately attend to the correct regions, which is evident from the high correctness score.

From Tables 1 and 2, it can be seen that ARNN<sup>~</sup> provides no significant advantage over its deterministic counterpart. This can be attributed to the datasets encouraging point estimates, as each query can only have one correct answer. As a consequence, for each  $a_{i,j}, \sigma_{i,j}$  was observed to underestimate variance. However, in situations where a query can have multiple correct answers, ARNN<sup>~</sup> can generate diverse attention masks. To corroborate this claim, we test the pre-trained ARNN<sup>~</sup> on images that are similar to the MBG dataset but have the same digit in multiple colors. Figure 6a shows the individual layer attended feature maps for three different samples from ARNN<sup>~</sup> for a fixed image and query. For the query “9”, ARNN<sup>~</sup> is able to identify the three modes. Note that since  $\sigma_{i,j}$ ’s were underestimated due to aforementioned reasons, they were scaled up before generating the samples. Despite being underestimated  $\sigma_{i,j}$ ’s still capture crucial information.

**Inverse Attribute Prediction.** Figure 6a leads to an interesting observation regarding the nature of the task. Even though ARNN<sup>~</sup> is able to identify the correct number, it

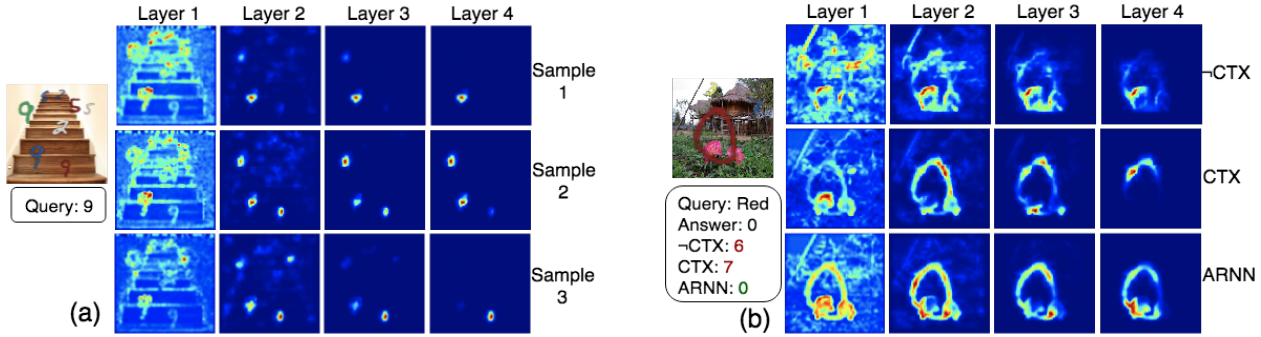


Figure 6: **Qualitative analysis of the attention masks.** (a) Layer-wise attended feature maps sampled from ARNN<sup>~</sup>. The samples span all the modes in the image. (b) Layer-wise attended feature maps generated by different mechanisms visualized on images from MBG<sup>inv</sup> dataset. Additional visualizations are shown in the supplementary material.

	Total	Scale				
		0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0
NONE	91.43	85.63	92.57	94.96	94.77	93.59
ARNN	91.09	84.89	92.25	94.24	94.70	<b>94.52</b>
BRNN <sup>γ=3</sup>	91.67	85.97	93.46	94.81	94.35	93.68
BRNN <sup>γ=2</sup>	<b>92.68</b>	<b>88.10</b>	<b>94.23</b>	<b>95.32</b>	<b>94.80</b>	94.01

Table 3: **Block AttentionRNN.** Ablation results on MBG<sup>b</sup> dataset. AttentionRNN (ARNN) and Block AttentionRNN (BRNN) with block sizes of 2 and 3 are compared.

only needs to focus on a tiny part of the target region to be able to accurately classify the color. To further demonstrate the ARNN’s ability to model longer dependencies, we test the performance of ARNN, CTX and  $\neg$ CTX on the MBG<sup>inv</sup> dataset, which defines the inverse attribute prediction problem - given a color identify the number corresponding to that color. The base CNN architecture is identical to the one used in the previous experiment. ARNN, CTX and  $\neg$ CTX achieve an accuracy of 72.77%, 66.37% and 40.15% and a correctness score [20] of 0.39, 0.24 and 0.20 respectively. Figure 6b shows layer-wise attended feature maps for the three models. ARNN is able to capture the entire number structure, whereas the other two methods only focus on a part of the target region. Even though CTX uses some local context to compute the attention masks, it fails to identify the complete structure for the number “0”. A plausible reason for this is that a  $3 \times 3$  local context is too small to capture the entire target region. As a consequence, the attention mask is computed in patches. CTX maintains no information about the previously computed attention values, and therefore is unable to assign correlated attention scores for all the different target region patches. ARNN, on the other hand, captures constraints between attention variables, making it much more effective in this situation.

**Scalability of ARNN.** The results shown in Table 1 correspond to models trained on  $100 \times 100$  input images, where the first attention layer is applied on an image feature of size  $50 \times 50$ . To analyze the performance of ARNN on comparatively larger image features, we create a new dataset of  $224 \times 224$  images which we refer to as MBG<sup>b</sup>. The data gen-

eration process for MBG<sup>b</sup> is identical to MBG. We perform an ablation study analyzing the effect of using Block AttentionRNN (BRNN) (Section 3.3) instead of ARNN on larger image features. For the base architecture, the ARNN model from the previous experiment is augmented with an additional stack of convolutional and max pooling layer. The detailed architecture is mentioned in the supplementary material. Table 3 shows the color prediction accuracy on different scale intervals for the MBG<sup>b</sup> dataset. As the first attention layer is now applied on a feature map of size  $112 \times 112$ , ARNN performs worse than the case when no attention (NONE) is applied due to the poor tractability of LSTMs over large sequences. BRNN<sup>γ=2</sup>, on the other hand, is able to perform better as it reduces the image feature size before applying attention. However, there is a considerable difference in the performance of BRNN when  $\gamma = 2$  and  $\gamma = 3$ . When  $\gamma = 3$ , BRNN applies a  $3 \times 3$  convolution with stride 3. This aggressive size reduction causes loss of information.

## 4.2. Image Classification

**Dataset.** We use the CIFAR-100 [16] to verify the performance of AttentionRNN on the task of image classification. The dataset consists of 60,000  $32 \times 32$  images from 100 classes. The training/test set contain 50,000/10,000 images.

**Experimental Setup.** We augment ARNN to the convolution block attention module (CBAM) proposed by [35]. For a given feature map, CBAM computes two different types of attentions: 1) channel attention that exploits the inter-channel dependencies in a feature map, and 2) spatial attention that uses local context to identify relationships in the spatial domain. We replace *only* the spatial attention in CBAM with ARNN. This modified module is referred to as CBAM+ARNN. ResNet18 [8] is used as the base model for our experiments. ResNet18+CBAM is the model obtained by using CBAM in the Resnet18 model, as described in [35]. Resnet18+CBAM+ARNN is defined analogously. We use a local context of  $3 \times 3$  to compute the spatial attention for both CBAM and CBAM+ARNN.

**Results.** Top-1 and top-5 error is used to evaluate the performance of the models. The results are summarized in Ta-



	Top-1 Error (%)	Top-5 Error (%)	Rel. Runtime
ResNet18 [8]	25.56	6.87	1x
ResNet18 + CBAM [35]	25.07	6.57	1.43x
ResNet18 + CBAM + ARNN	<b>24.18</b>	<b>6.42</b>	4.81x

Table 4: **Performance on Image Classification.** The Top-1 and Top-5 error % are shown for all the models. The ARNN based model outperforms all other baselines.

	Yes/No	Number	Other	Total	Rel. Runtime
MCB [5]	76.06	35.32	43.87	54.84	1x
MCB+ATT [5]	76.12	35.84	47.84	56.89	1.66x
MCB+ARNN	<b>77.13</b>	<b>36.75</b>	<b>48.23</b>	<b>57.58</b>	2.46x

Table 5: **Performance on VQA.** In % accuracy.

ble 4. CBAM+ARNN provides an improvement of 0.89% on top-1 error over the closest baseline. Note that this gain, though seems marginal, is larger than what CBAM obtains over ResNet18 with no attention (0.49% on top-1 error).

### 4.3. Visual Question Answering

**Dataset.** We evaluate the performance of ARNN on the task of VQA [2]. The experiments are done on the VQA 2.0 dataset [7], which contains images from MSCOCO [19] and corresponding questions. As the test set is not publicly available, we evaluate performance on the validation set.

**Experimental Setup.** We augment ARNN to the Multi-modal Compact Bilinear Pooling (MCB) architecture proposed by [5]. This is referred to as MCB+ARNN. Note that even though MCB doesn’t give state-of-the-art performance on this task, it is a competitive baseline that allows for easy ARNN integration. MCB+ATT is a variant to MCB that uses a local attention mechanism with  $\delta = 1$  from [5]. For fair comparison, MCB+ARNN also uses a  $\delta = 1$  context.

**Results.** The models are evaluated using the accuracy measure defined in [2]. The results are summarized in Table 5. MCB+ARNN achieves a 0.69% improvement over the closest baseline<sup>1</sup>. We believe this marginal improvement is because all the models, for each spatial location  $(i, j)$ , use no context from neighbouring locations (as  $\delta = 1$ ).

### 4.4. Image Generation

**Dataset.** We analyze the effect of using ARNN on the task of image generation. Experiments are performed on the CelebA dataset [21], which contains 202,599 face images of celebrities, with 40 binary attributes. The data pre-processing is identical to [43]. The models are evaluated on three attributes: hair color = {*black*, *blond*, *brown*}, gender = {*male*, *female*}, and smile = {*smile*, *nosmile*}.

<sup>1</sup>Reported performance for the baselines are from our PyTorch [23] re-implementation of [5], and are directly comparable to MCB+ARNN. We note that [7] reports slightly better performance (by  $\sim 2\%$ ), which we attribute to differences in frameworks (PyTorch [23] vs Caffe [12]) and possible use of additional features (as noted in Table 4 of [5]).

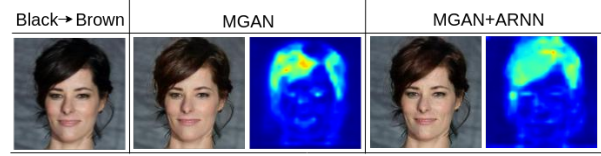


Figure 7: **Qualitative Results on ModularGAN.** Attention masks generated by original ModularGAN [43] and ModularGAN augmented with ARNN are shown. Notice that the hair mask is more uniform for MGAN+ARNN as it is able to encode structural dependencies in the attention mask. Additional results shown in supplementary material.

	Hair	Gender	Smile	Rel. Runtime
MGAN [43]	<b>2.5</b>	3.2	12.6	1x
MGAN+ARNN	3.0	<b>1.4</b>	<b>11.4</b>	1.96x

Table 6: **Performance on Image Generation.** ResNet18 [8] Classification Errors (%) for each attribute transformation. ARNN achieves better performance on two tasks.

**Experimental Setup.** We compare ARNN to a local attention mechanism used in the ModularGAN (MGAN) framework [43]. MGAN uses a  $3 \times 3$  local context to obtain attention values. We define MGAN+ARNN as the network obtained by replacing the local attention with ARNN. The models are trained to transform an image given an attribute.

**Results.** To evaluate the performance of the models, similar to [43], we train a ResNet18 [8] model that classifies the hair color, facial expression and gender on the CelebA dataset. The trained classifier achieves an accuracy of 93.9%, 99.0% and 93.7% on hair color, gender and smile respectively. For each transformation, we pass the generated images through this classifier and compute the classification error (shown in Table 6). MGAN+ARNN outperforms the baseline on all categories except *hair color*. To analyse this further, we look at the attention masks generated for the *hair color* transformation by both models. As shown in Figure 7, we observe that the attention masks generated by MGAN lack coherence over the target region due to discontinuities. MGAN+ARNN, though has a slightly higher classification error, generates uniform activation values over the target region by encoding structural dependencies.

## 5. Conclusion

In this paper, we developed a novel *structured* spatial attention mechanism which is end-to-end trainable and can be integrated with any feed-forward convolutional neural network. The proposed AttentionRNN layer explicitly enforces structure over the spatial attention variables by sequentially predicting attention values in the spatial mask. Experiments show consistent quantitative and qualitative improvements on a large variety of recognition tasks.

**Acknowledgments:** This work was funded in part by the Vector Institute for AI, Canada CIFAR AI Chair, NSERC CRC and an NSERC DG and Discovery Accelerator Grants.



## References

- [1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2
- [2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 1, 8
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015. 3
- [4] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6298–6306. IEEE, 2017. 1, 2
- [5] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Conference on Empirical Methods in Natural Language Processing*, pages 457–468. ACL, 2016. 1, 2, 3, 8
- [6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 3
- [7] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6325–6334, 2017. 8
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 7, 8
- [9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3
- [10] Ronghang Hu, Jacob Andreas, Kate Saenko, and Trevor Darrell. Explainable neural computation via stack neural module networks. In *European Conference on Computer Vision*, 2018. 2
- [11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. 2
- [12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 8
- [13] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Fei-Fei Li, Larry Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1
- [14] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1
- [15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *stat*, 1050:10, 2014. 3
- [16] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 7
- [17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1
- [18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5
- [19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755, 2014. 8
- [20] Chenxi Liu, Junhua Mao, Fei Sha, and Alan L Yuille. Attention correctness in neural image captioning. In *AAAI*, pages 4176–4182, 2017. 6, 7
- [21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, 2015. 8
- [22] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016. 1, 2, 3
- [23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 8
- [24] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *International Conference on Machine Learning*, 2014. 3
- [25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4
- [26] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *ICLR*, 2017. 3
- [27] Paul Hongsuck Seo, Andreas Lehrmann, Bohyung Han, and Leonid Sigal. Visual reference resolution using attention memory for visual dialog. In *Advances in Neural Information Processing Systems*, pages 3719–3729, 2017. 1, 2, 3, 6
- [28] Paul Hongsuck Seo, Zhe Lin, Scott Cohen, Xiaohui Shen, and Bohyung Han. Progressive attention networks for visual attribute prediction. *British Machine Vision Conference*, 2018. 1, 2, 3, 5, 6

- [29] Shiv Shankar and Sunita Sarawagi. Posterior attention models for sequence to sequence learning. In *International Conference on Learning Representations*, 2019. 3
- [30] Yichuan Tang, Nitish Srivastava, and Ruslan R. Salakhutdinov. Learning generative models with visual attention. In *Advances in Neural Information Processing Systems*, 2014. 1
- [31] Tatiana Tommas, Arun Mallya, Bryan Plummer, Svetlana Lazebnik, Alexander Berg, and Tamara Berg. Solving visual madlibs with multiple cues. In *British Machine Vision Conference*, 2016. 1
- [32] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016. 3
- [33] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *IEEE International Conference on Machine Learning*, pages 1747–1756, 2016. 2, 3, 4, 5
- [34] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015. 3
- [35] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *European Conference on Computer Vision*, pages 3–19, 2018. 1, 2, 3, 7, 8
- [36] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119(1):3–22, 2016. 6
- [37] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466, 2016. 1, 2
- [38] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 1, 2, 3, 6
- [39] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016. 1, 2
- [40] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016. 1, 2
- [41] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *arXiv preprint arXiv:1805.08318*, 2018. 1
- [42] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *IEEE International Conference on Computer Vision*, 2017. 3
- [43] Bo Zhao, Bo Chang, Zequn Jie, and Leonid Sigal. Modular generative adversarial networks. *European Conference on Computer Vision*, 2018. 1, 2, 3, 8
- [44] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE International Conference on Computer Vision*, volume 6, 2017. 1, 2
- [45] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004, 2016. 1, 2, 3