Supplementary Material for "ATTENTIONRNN: A Structured Spatial Attention Mechanism"

Siddhesh Khandelwal[1,2] and Leonid Sigal[1,2,3]
[1]University of British Columbia [2]Vector Institute for AI [3]Canada CIFAR AI Chair
{skhandel, lsigal}@cs.ubc.ca

Section 1 comments on the use of more expressive architectures for local spatial context computation (Section 3 in the main paper). Section 2 explains the architectures for the models used in the experiments (Section 4 in the main paper). Section 3 provides additional visualizations for the task of Visual Attribute Prediction (Section 4.1 in the main paper) and Image Generation (Section 4.4 in the main paper), further showing the effectiveness of our proposed structured attention mechanism.

1. Employing More Expressive Local Spatial Context δ

When computing the attention $a_{i,j}$ for the spatial location (i, j), as explained in Section 3 of the main paper, the proposed AttentionRNN (ARNN) mechanism utilizes local spatial context $\delta(\mathbf{x}_{i,j})$ in its formulation as a proxy for image features. In our approach, for computational simplicity and to ensure fair comparison to other attention mechanisms, this spatial context is realized using a single convolutional kernel (Eq. 9 in the main paper). We would like to highlight our proposed mechanism imposes no constraints on how the spatial context δ is modelled. One can easily use more complex networks $\mathcal{N}$ (such as U-Net [3]) to emulate $\delta(\mathbf{x}_{i,j})$. More specifically, only Eq. 9 of the main paper needs to be modified to accommodate this change.

$$\widehat{\mathbf{X}}^c = \text{skew}\left(\mathcal{N}(\mathbf{X})\right) \tag{1}$$

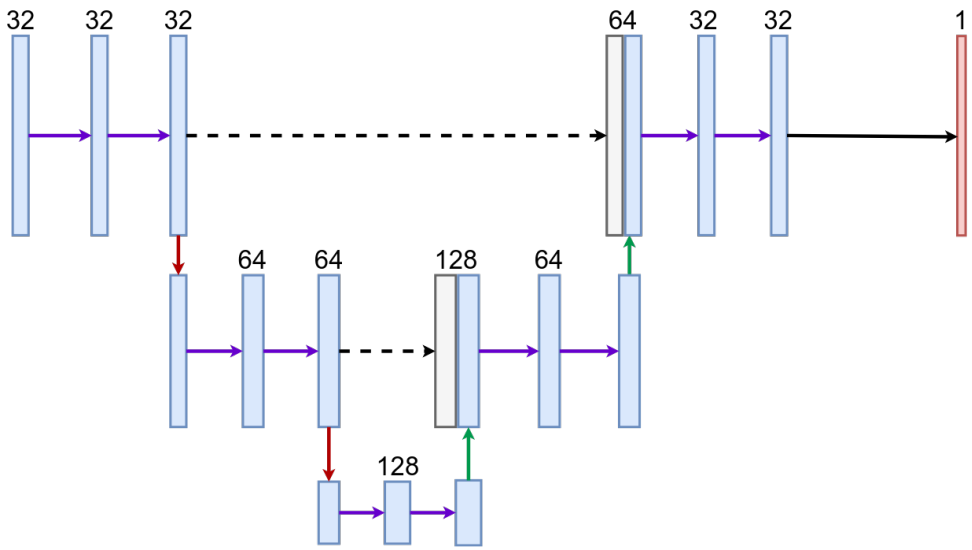
where $\mathcal{N}(\cdot)$ implies passing the input through the network $\mathcal{N}$.

Note that any network $\mathcal{N}$ used to realize the δ can also be used to generate a valid attention mask $\mathbf{A}$ by slightly modifying the output of $\mathcal{N}$ (for example applying a sigmoid non-linearity). Therefore $\mathcal{N}$ can be thought of as a local attention mechanism, as it uses local image information to compute $\mathbf{A}$. We would like to emphasize that our proposed AttentionRNN mechanism is complementary to any local attention mechanism $\mathcal{N}$, as one can easily incorporate $\mathcal{N}$ into the ARNN formulation as described earlier (Eq 1). Using ARNN in conjunction to $\mathcal{N}$ helps capture *explicit* global constraints over the attention variables, which, albeit at the cost of speed, can provide significant performance gains.

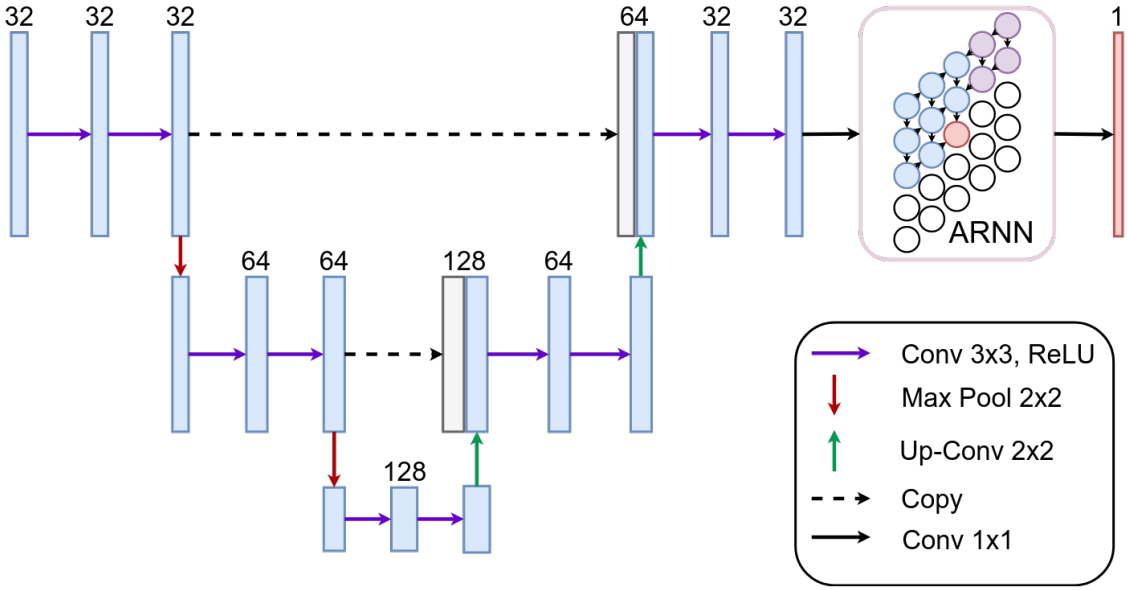
To further illustrate this, Table 1 compares the performance of using ARNN in conjunction with U-Net [3] on the MBG dataset. Please refer to Section 4.1 of the main paper for details on the dataset. All models share the same base network architecture (Table 2) barring the type of attention used. UNet$^{d=i}$ implies the use of a depth i U-Net architecture [3] to compute attention at each layer. Figure 1 highlights the differences between the attention layers used in UNet$^{d=i}$ and UNet$^{d=i}$+ARNN for $d = 3$. The differences for $d = 2$ are analogous. It can be seen that the use of ARNN to model structural dependencies significantly improves the performance ($\sim 5\%$).

		Scale				
	Total	0.5-1.0	1.0-1.5	1.5-2.0	2.0-2.5	2.5-3.0
UNet$^{d=2}$[3]	85.86	84.25	91.28	89.84	85.41	86.44
UNet$^{d=3}$ [3]	86.64	84.52	92.98	91.35	91.96	86.44
UNet$^{d=2}$+ARNN	91.73	90.47	95.78	93.98	92.26	93.89
UNet$^{d=3}$+ARNN	**92.29**	**90.86**	**96.14**	**94.54**	**97.02**	**94.91**

Table 1: **Color prediction accuracy on MBG dataset.** Results are in %. Using ARNN is conjunction to local attention mechanisms significantly improves performance.



(a) Attention layer used in UNet$^{d=3}$



(b) Attention layer used in UNet$^{d=3}$+ARNN

Figure 1: **Difference between the attention layers used in UNet$^{d=3}$ and UNet$^{d=3}$+ARNN.** Both UNet$^{d=3}$ and UNet$^{d=3}$+ARNN use the same base architecture (Table 2) barring the attention layer. (a) The attention layer used after each pooling operation in UNet$^{d=3}$. (b) The attention layer used after each pooling operation in UNet$^{d=3}$+ARNN. The details regarding the incorporation of UNet [3] output into ARNN is explained in Section 1.

2. Model Architectures

2.1. Visual Attribute Prediction

Please refer to Section 4.1 of the main paper for the task definition. Similar to [4], the base CNN architecture is composed of four stacks of 3×3 convolutions with 32 channels followed by 2×2 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. Table 2 illustrates the model architectures for each network. $\{\neg\text{CTX, CTX, ARNN}\}_{sigmoid}$ refers to using sigmoid non-linearity on the generated attention mask before applying it to the image features. Similarly, $\{\neg\text{CTX, CTX, ARNN}\}_{softmax}$ refers to using softmax non-linearity on the generated attention mask. We use the same hyper-parameters and training procedure for all models, which is identical to [4].

For the scalability experiment described in Section 4.1, we add an additional stack of 3×3 convolution layer followed by a 2×2 max pooling layer to the ARNN architecture described in Table 2. This is used as the base architecture. Table 3 illustrates the differences between the models used to obtain results mentioned in Table 3 of the main paper.

SAN	**¬CTX**	**CTX**	**ARNN**
conv1 (3x3@32)			
pool1 (2x2)			
↓	$\neg\text{CTX}_{sigmoid}$	$\text{CTX}_{sigmoid}$	$\text{ARNN}_{sigmoid}$
conv2 (3x3@32)			
pool2 (2x2)			
↓	$\neg\text{CTX}_{sigmoid}$	$\text{CTX}_{sigmoid}$	$\text{ARNN}_{sigmoid}$
conv3 (3x3@32)			
pool3 (2x2)			
↓	$\neg\text{CTX}_{sigmoid}$	$\text{CTX}_{sigmoid}$	$\text{ARNN}_{sigmoid}$
conv4 (3x3@32)			
pool4 (2x2)			
SAN	$\neg\text{CTX}_{softmax}$	$\text{CTX}_{softmax}$	$\text{ARNN}_{softmax}$

Table 2: Architectures for the models used in Section 4.1 of the main paper. ↓ implies that the previous and the next layer are directly connected. The input is passed to the top-most layer. The computation proceeds from top to bottom.

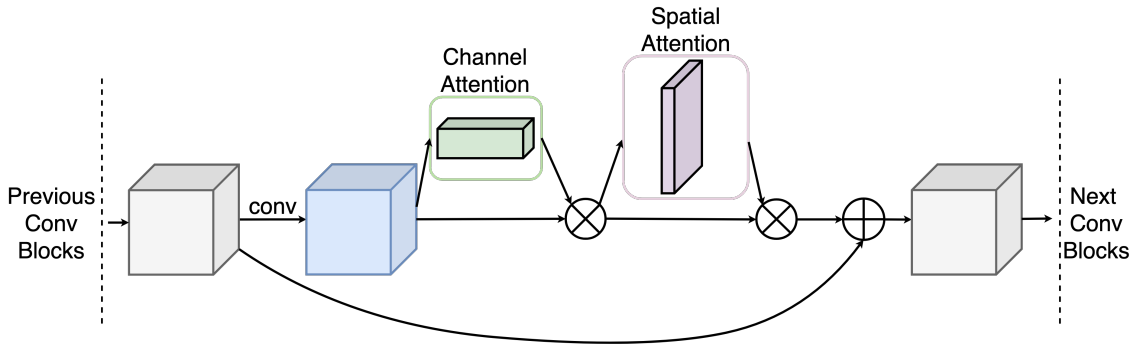
NONE	**ARNN**	**BRNN**
conv1 (3x3@32)		
pool1 (2x2)		
↓	$\text{ARNN}_{sigmoid}$	$\text{BRNN}^{\delta}_{sigmoid}$
ARNN (described in Table 2)		

Table 3: Model architectures for the scalability study described in Section 4.1 of the main paper. ↓ implies that the previous and the next layer are directly connected. **ARRN** in defined in Table 2.

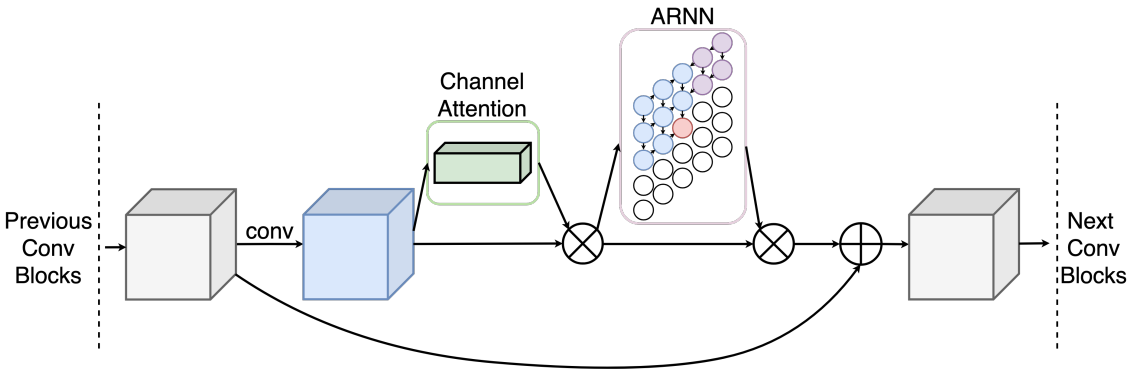
2.2. Image Classification

Please refer to Section 4.2 of the main paper for the task definition. We augment ARNN to the convolution block attention module (CBAM) proposed by [5]. 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. Figure 2a shows the CBAM module integrated with a ResNet [2] block. We replace only the *spatial attention* in CBAM with ARNN. This modified module is referred to as CBAM+ARNN. Figure 2b better illustrates this modification. Both CBAM and CBAM+ARNN use a local context of 3×3 to compute attention. We use the same hyper-parameters and training procedure for both CBAM and CBAM+ARNN, which is identical to [5].



(a) CBAM module

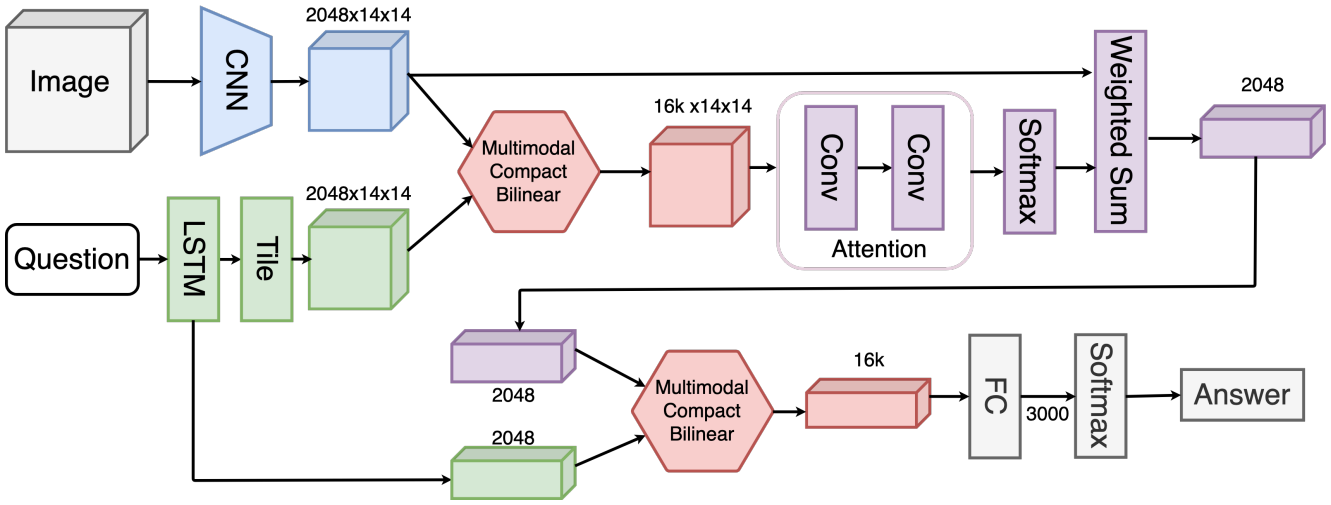


(b) CBAM+ARNN module

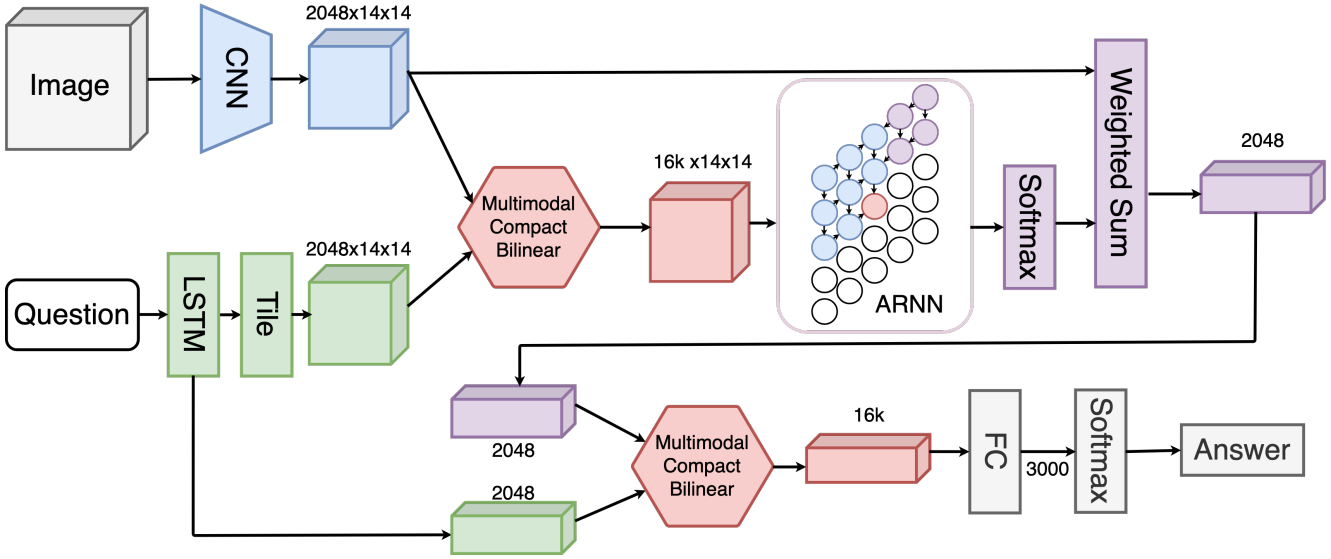
Figure 2: **Difference between CBAM and CBAM+ARNN.** (a) CBAM[5] module integrated with a ResNet[2] block. (b) CBAM+ARNN replaces the spatial attention in CBAM with ARNN. It is applied similar to (a) after each ResNet[2] block. Refer to Section 4.2 of the main paper for more details.

2.3. Visual Question Answering

Please refer to Section 4.3 of the main paper for task definition. We use the Multimodal Compact Bilinear Pooling with Attention (MCB+ATT) architecture proposed by [1] as a baseline for our experiment. To compute attention, MCB+ATT uses two 1×1 convolutions over the features obtained after using the compact bilinear pooling operation. Figure 3a illustrates the architecture for MCB+ATT. We replace this attention with ARNN to obtain MCB+ARNN. MCB+ARNN also uses a 1×1 local context to compute attention. Figure 3b better illustrates this modification. We use the same hyper-parameters and training procedure for MCB, MCB+ATT and MCB+ARNN, which is identical to [1].



(a) MCB+ATT



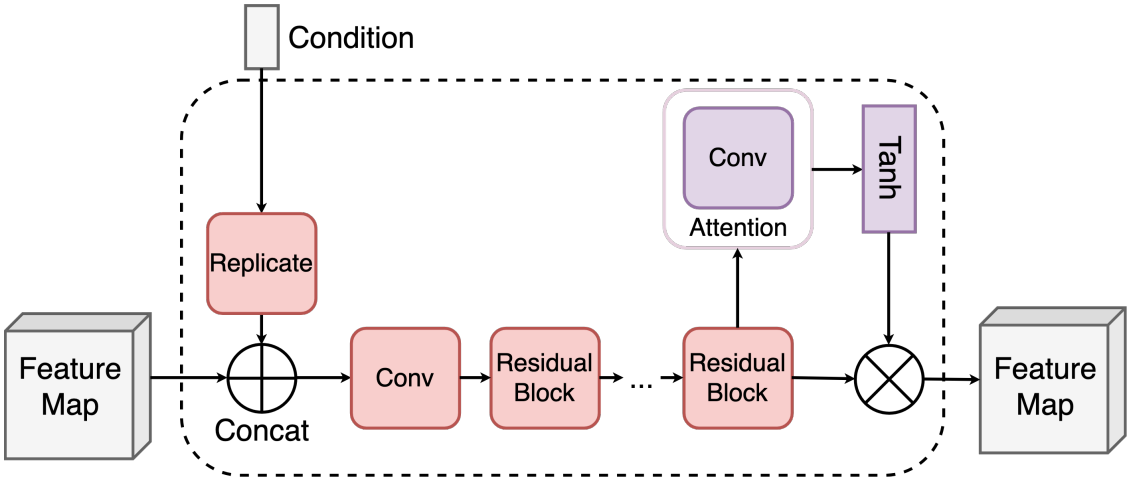
(b) MCB+ARNN

Figure 3: **Difference between MCB+ATT and MCB+ARNN.** (a) MCB+ATT model architecture proposed by [1]. It uses a 1×1 context to compute attention over the image features. (b) MCB+ARNN replaces the attention mechanism in MCB+ATT with ARNN. It is applied in the same location as (a) with 1×1 context. Refer to Section 4.3 of the main paper for more details.

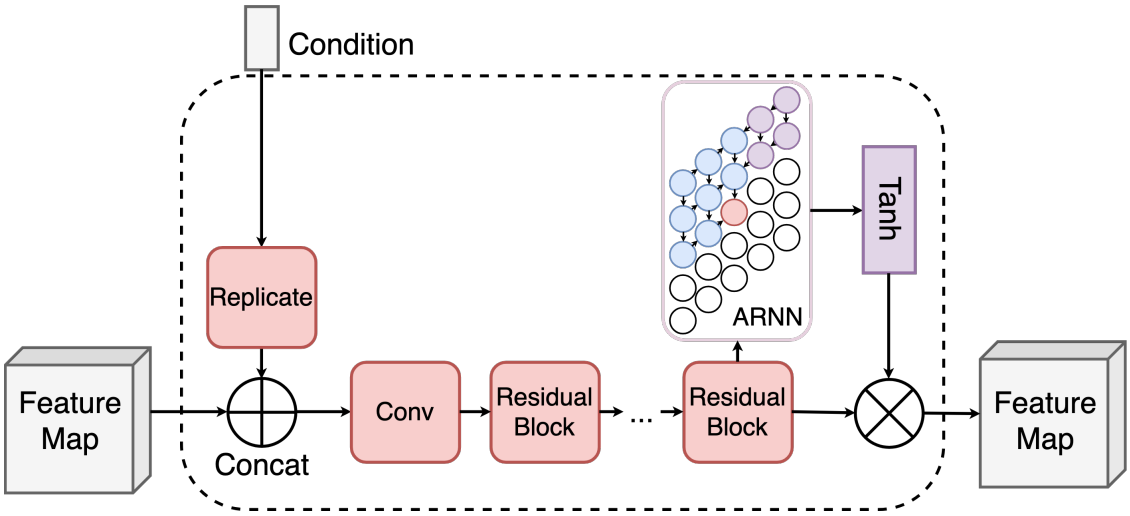
2.4. Image Generation

Please refer to Section 4.4 of the main paper for task definitions. We compare ARNN to a local attention mechanism used in the ModularGAN (MGAN) framework [6]. MGAN consists of three modules: 1) encoder module that encodes an input image into an intermediate feature representation, 2) generator module that generates an image given an intermediate feature representation as input, and 3) transformer module that transforms a given intermediate representation to a new intermediate representation according to some input condition. The transformer module uses a 3×3 local context to compute attention over the feature representations. Figure 4a illustrates the transformer module proposed by [6]. We define MGAN+ARNN as the network obtained by replacing this local attention mechanism in the transformer module with ARNN. Note that the generator and encoder modules are unchanged. MGAN+ARNN also uses a 3×3 local context to compute attention. Figure 4b better illustrates this modification to the transformer module. We use the same hyper-parameters and training procedure

for both MGAN and MGAN+ARNN, which is identical to [6].



(a) Transformer module for MGAN



(b) Transformer module for MGAN+ARNN

Figure 4: **Difference between MGAN and MGAN+ARNN.** (a) The transformer module for the ModularGAN (MGAN) architecture proposed by [6]. It uses a 3×3 local context to compute attention over the intermediate features. (b) MGAN+ARNN replaces the attention mechanism in MGAN with ARNN. It is applied in the same location as (a) with 3×3 local context. Note that the generator and encoder modules in MGAN and MGAN+ARNN are identical. Refer to Section 4.4 of the main paper for more details.

3. Additional Visualizations

3.1. Visual Attribute Prediction

Please refer to Section 4.1 of the main paper for task definition. Figures 5 - 7 show the individual layer attended feature maps for three different samples from ARNN$^{\sim}$ for a fixed image and query. It can be seen that ARNN$^{\sim}$ is able to identify the different modes in each of the images.

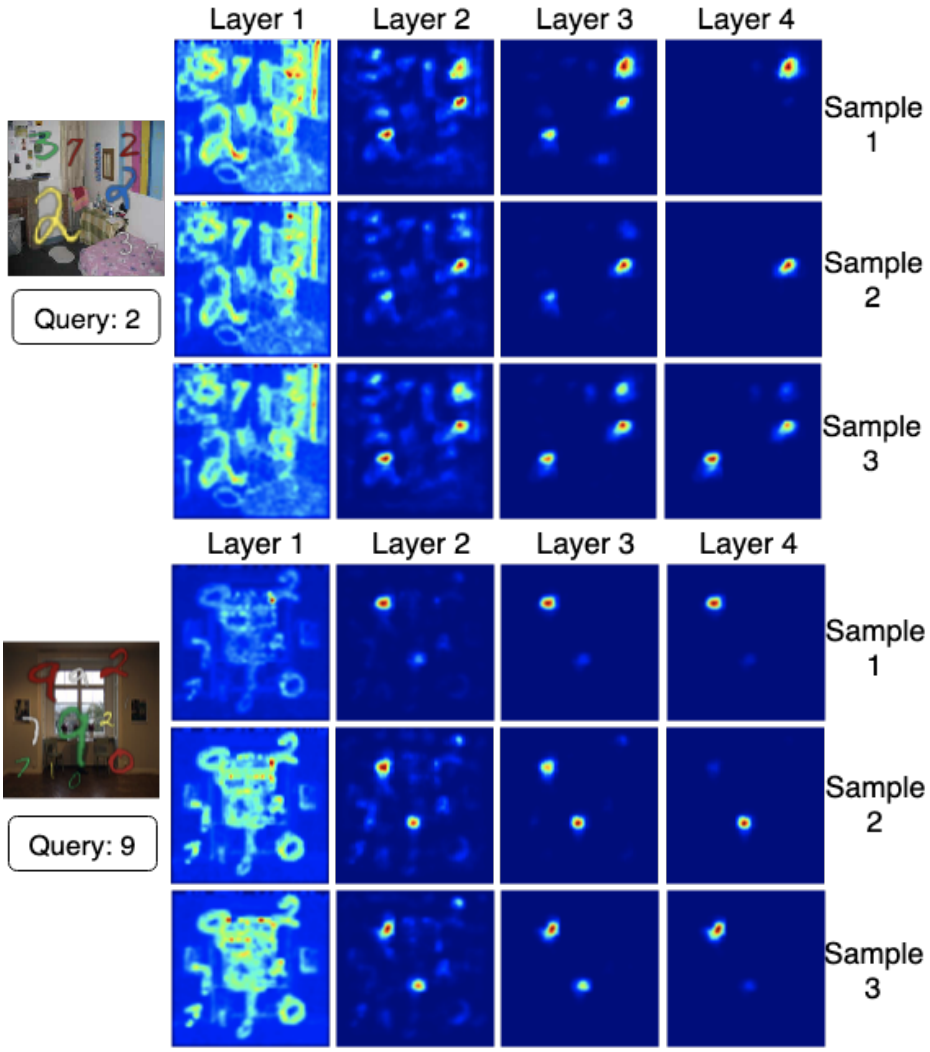


Figure 5: **Qualitative Analysis of Attention Masks sampled from ARNN$^{\sim}$.** Layer-wise attended feature maps sampled from ARNN$^{\sim}$ for a fixed image and query. The masks are able to span the different modes in the image. For detailed explanation see Section 4.1 of the main paper.

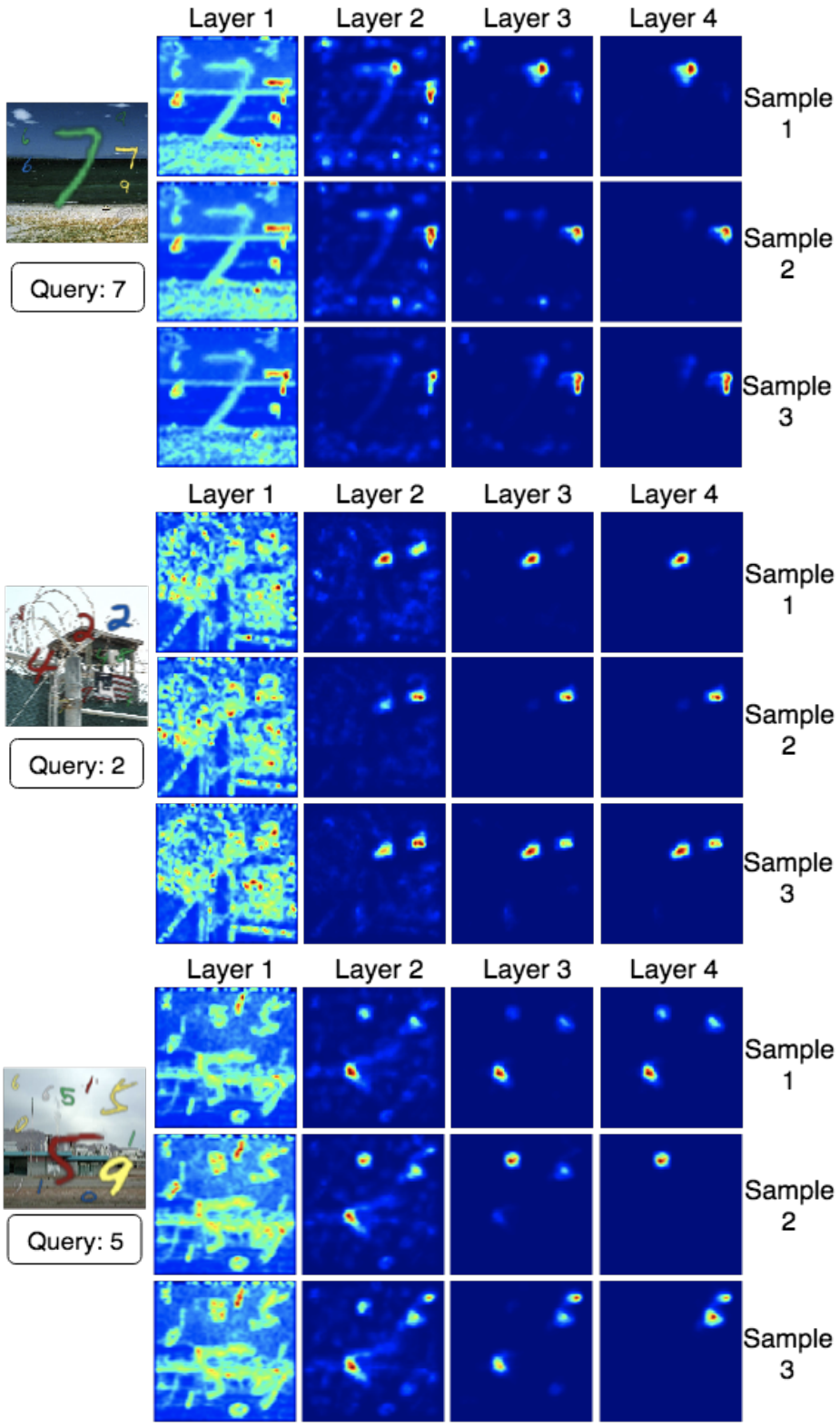


Figure 6: **Qualitative Analysis of Attention Masks sampled from ARNN$^{\sim}$.** Layer-wise attended feature maps sampled from ARNN$^{\sim}$ for a fixed image and query. The masks are able to span the different modes in the image. For detailed explanation see Section 4.1 of the main paper.

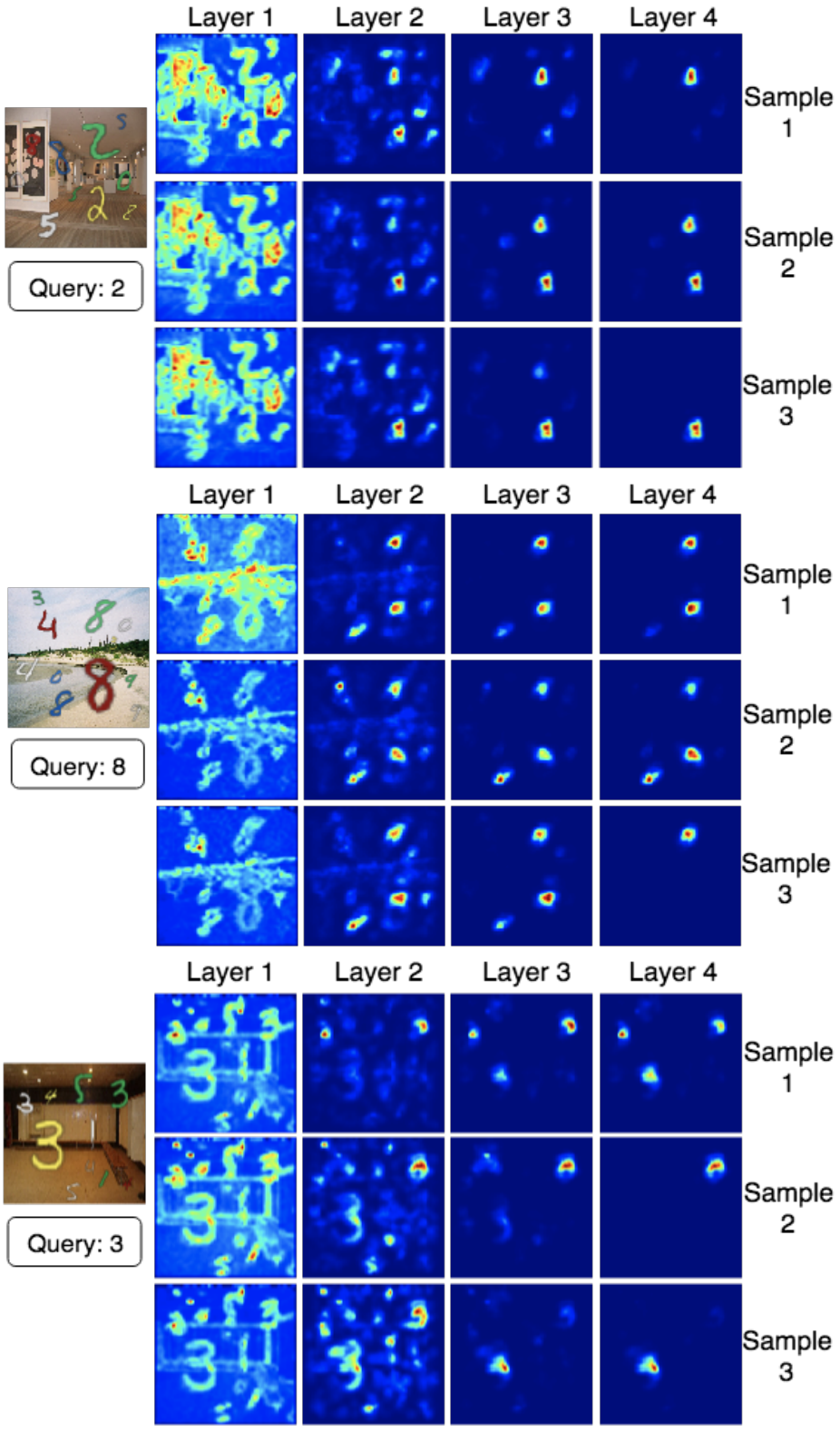


Figure 7: **Qualitative Analysis of Attention Masks sampled from ARNN$^{\sim}$.** Layer-wise attended feature maps sampled from ARNN$^{\sim}$ for a fixed image and query. The masks are able to span the different modes in the image. For detailed explanation see Section 4.1 of the main paper.

3.2. Inverse Attribute Prediction

Please refer to Section 4.1 of the main paper for task definition. Figures 8 - 10 show the individual layer attended feature maps comparing the different attention mechanisms on the MBG^{inv} dataset. It can be seen that ARNN captures the entire number structure, whereas the other two methods only focus on a part of the target region or on some background region with the same color as the number, leading to incorrect predictions.

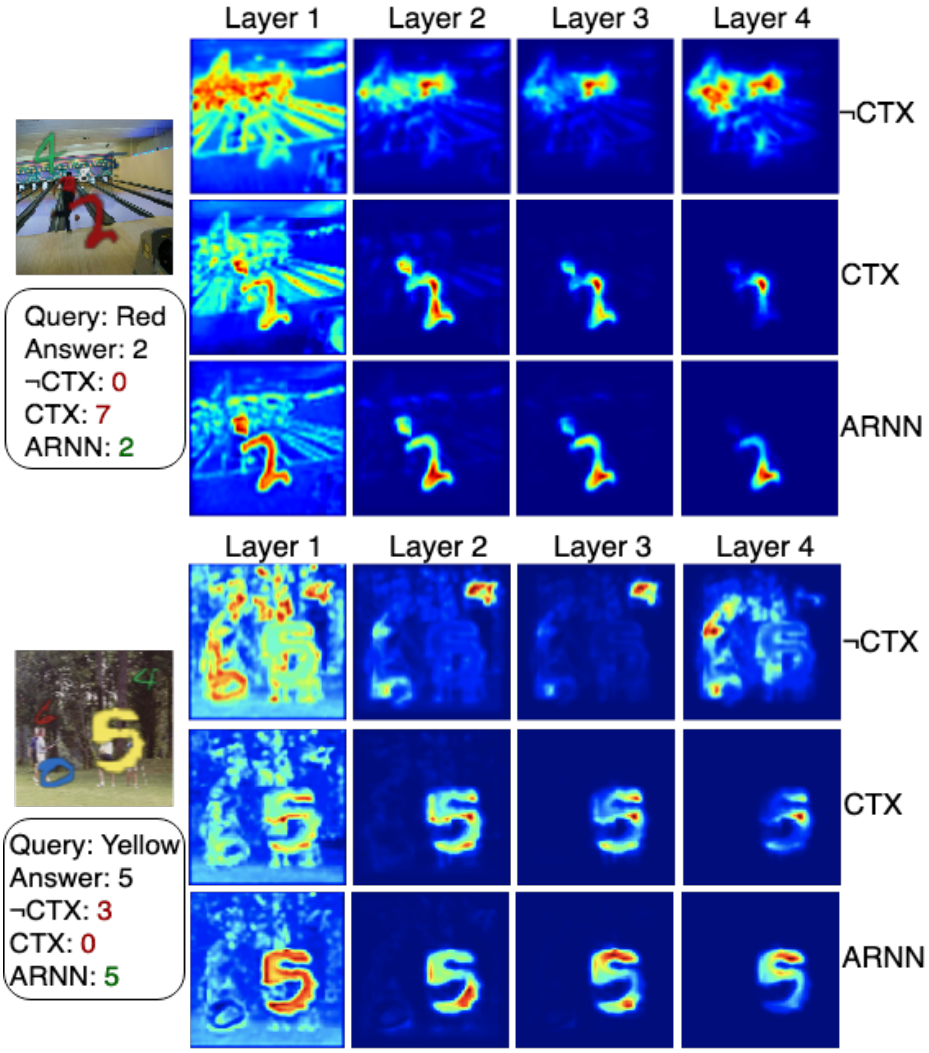


Figure 8: **Qualitative Analysis of Attention Masks on MBG^{inv}.** Layer-wise attended feature maps generated by different mechanisms visualized on images from MBG^{inv} dataset. ARNN is able to capture the entire number structure, whereas the other two methods only focus on a part of the target region or on some background region with the same color as the target number. For detailed explanation see Section 4.1 of the main paper.

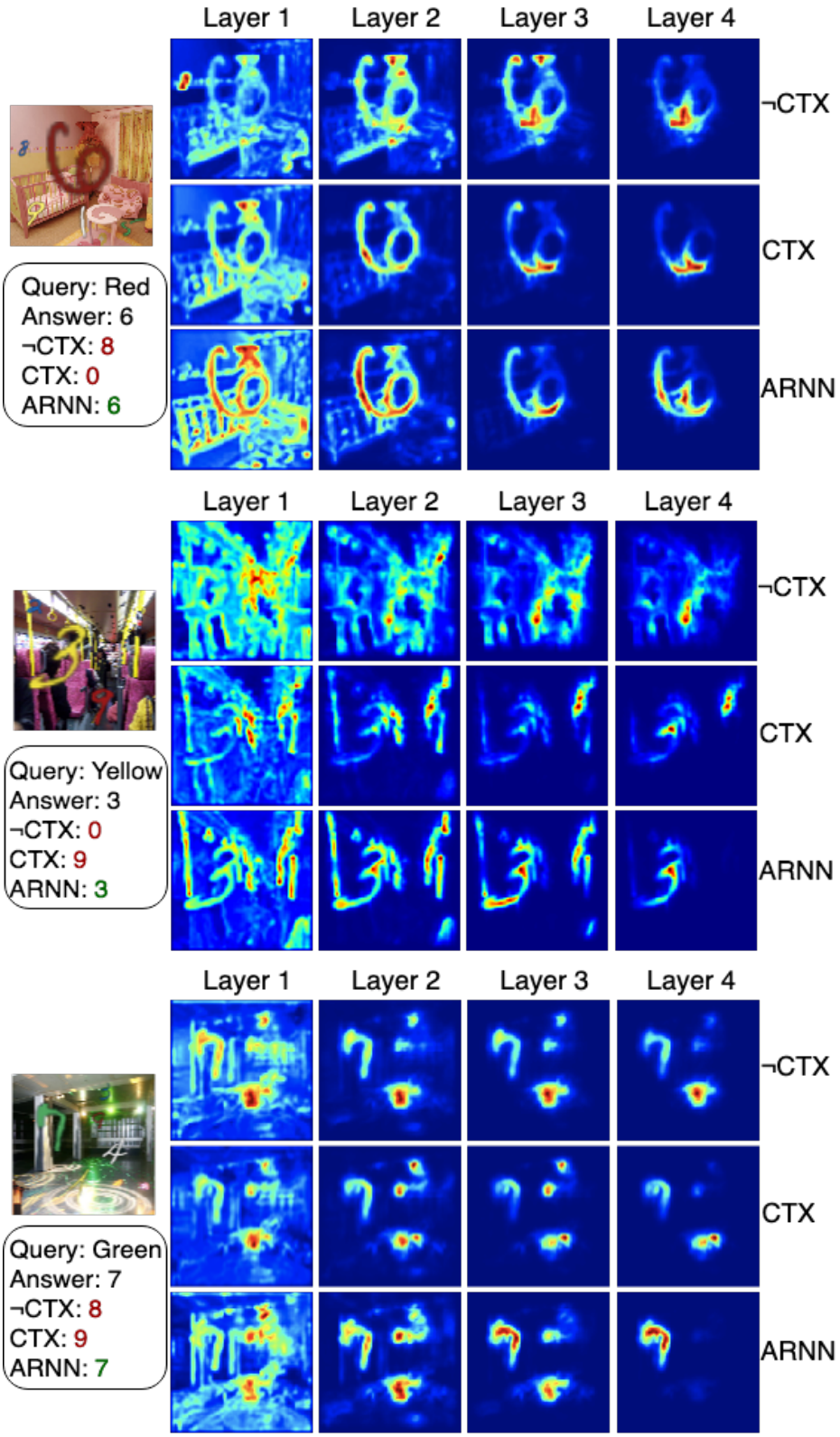


Figure 9: **Qualitative Analysis of Attention Masks on MBG^{inv}.** Layer-wise attended feature maps generated by different mechanisms visualized on images from MBG^{inv} dataset. ARNN is able to capture the entire number structure, whereas the other two methods only focus on a part of the target region or on some background region with the same color as the target number. For detailed explanation see Section 4.1 of the main paper.

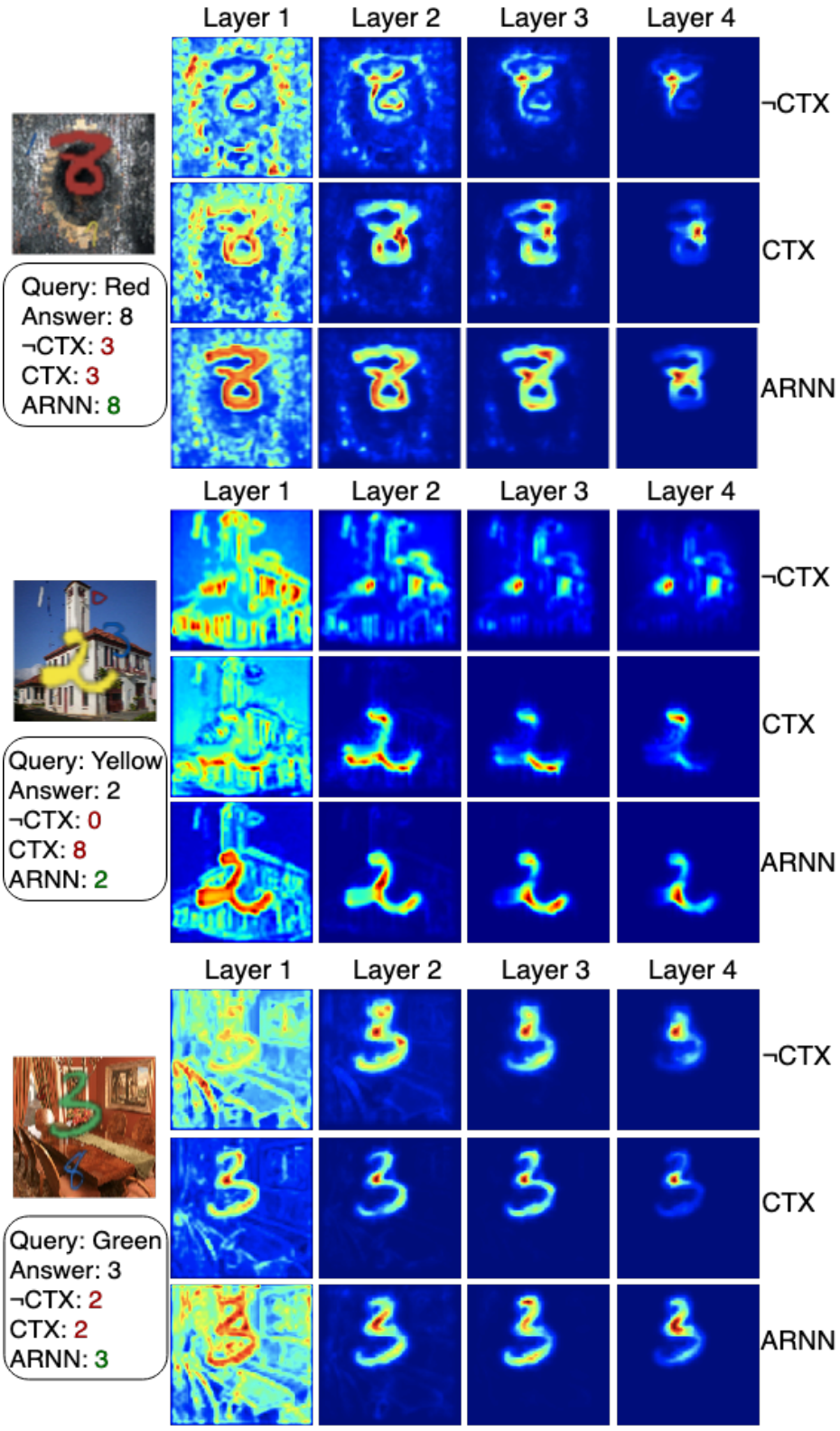


Figure 10: **Qualitative Analysis of Attention Masks on MBG^{inv}.** Layer-wise attended feature maps generated by different mechanisms visualized on images from MBG^{inv} dataset. ARNN is able to capture the entire number structure, whereas the other two methods only focus on a part of the target region or on some background region with the same color as the target number. For detailed explanation see Section 4.1 of the main paper.

3.3. Image Generation

Please refer to Section 4.4 of the main paper for task definition. Figures 11 and 12 show the attention masks generated by MGAN and MGAN+ARNN for the task of *hair color* transformation. MGAN+ARNN encodes structural dependencies in the attention values, which is evident from the more uniform and continuous attention masks. MGAN, on the other hand, has sharp discontinuities which, in some cases, leads to less accurate hair color transformations.

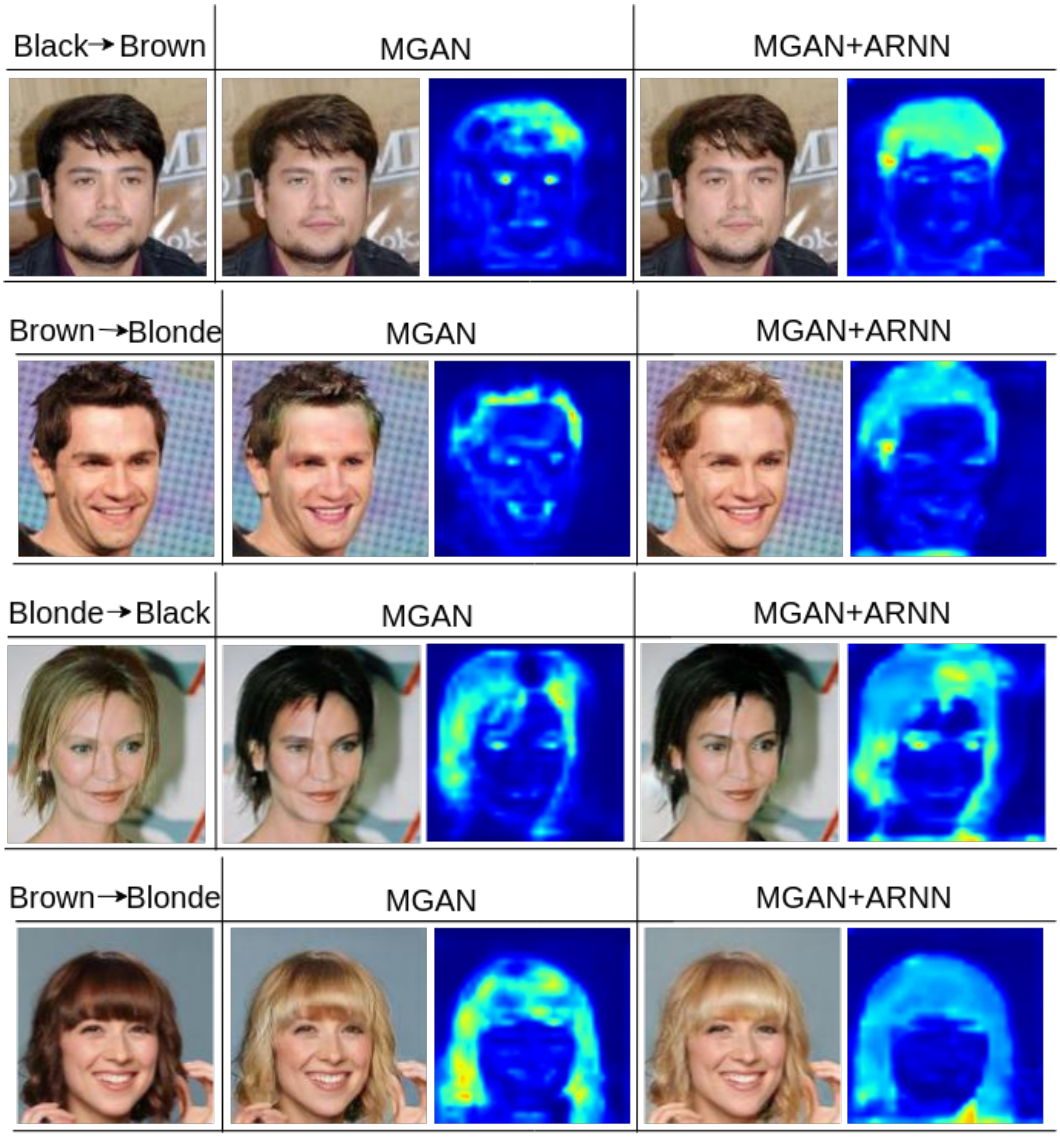


Figure 11: **Qualitative Results for Image Generation.** Attention masks generated by MGAN and MGAN+ARNN are shown. Notice that the hair mask is more uniform for MGAN+ARNN as it is able to encode structural dependencies in the attention mask. For detailed explanation see Section 4.4 of the main paper.

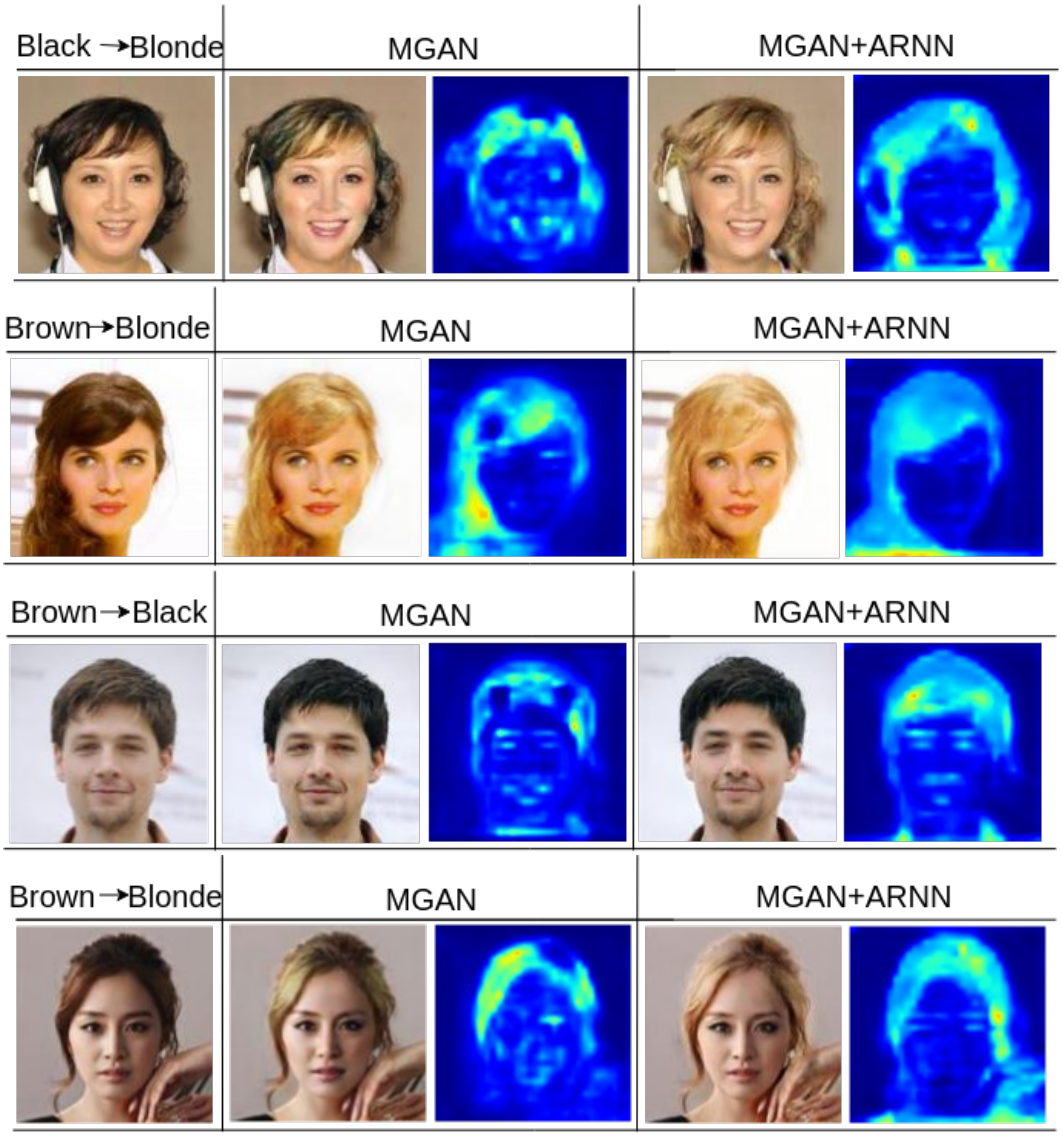


Figure 12: **Qualitative Results for Image Generation.** Attention masks generated by MGAN and MGAN+ARNN are shown. Notice that the hair mask is more uniform for MGAN+ARNN as it is able to encode structural dependencies in the attention mask. For detailed explanation see Section 4.4 of the main paper.

References

[1] 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. 4, 5

[2] 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. 4

[3] 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. 1, 2

[4] Paul Hongsuck Seo, Zhe Lin, Scott Cohen, Xiaohui Shen, and Bohyung Han. Progressive attention networks for visual attribute prediction. *British Machine Vision Conference*, 2018. 3

[5] 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. 3, 4

[6] Bo Zhao, Bo Chang, Zequn Jie, and Leonid Sigal. Modular generative adversarial networks. *European Conference on Computer Vision*, 2018. 5, 6