ISSN No:-2456-2165

Dense Caption Imagining

Vatsal Verma[1], Darpan Khanna[2], Gaurvi Vishnoi[3], and Shreyas Raturi[4]
Department of Computer Science and Engineering, College of Engineering and Technology,
Faculty of Engineering and Technology, SRM Institute of Science and Technology

Abstract:- A lot of recent research has focused on both computer vision and natural language processing. Our research focuses on the intersection of these, specifically generating pictures from captions. We focus on the lower data regime, using the COCO and CUB data sets which include 200k and 11k picture and caption pairs (respectively). We will use a hierarchical GAN architecture as our baseline[7][24][26]. To improve our baseline we attempt various methods targeting the upsampling blocks, and adding residual or attention-based layers. We will compare the inception score of the methods to analyze our results. We will also consider qualitative results to assure there is minimal mode collapse and memorization. We find that of all our improvements, improving the up-sampling technique to use a Laplacian pyramid method with transposed convolutional layers obtains the best results with a minimal increase in computation time and memory needs.

Keywords:- *Computer Vision, Natural Language Processing, stackGAN, Image Captioning, Machine Learning, Deep Learning.*

I. INTRODUCTION

A. *Text and Image Modeling*

Caption generation from images has been a relatively recent research focus, with many promising advances [9][8][19][21]. We will focus on creating images from captions. One of the earliest known attempts at this drew in patches onto a canvas using an attention-based language model and a generative RNN [12]. GANs quickly became one of the most popular methods. The first application of GANs used a DC-GAN that was conditioned on the caption encoding [14]. Related attempts used a similar text conditioning concept, but with an additional classification auxiliary loss [2]. Research into applying hierarchical GANs made progress in performance. StackGAN was one of the first attempts to do this. It has an initial layer that generates a simple, low-resolution image, whereas the second layer generates a more detailed and high-resolution image [24]. Notably, StackGAN used c-GANs [13]for both layers and introduced conditioning augmentation to handle discontinuity in the latent text space [24]. StackGAN++ improved onthis framework using multiple generative and discriminative layers in a tree [26]. Additional improvements have been made in this task by introducing new concepts [4]. By including attention to the stacked GAN architecture the generative process weighs important elements of the caption more [22]. Siamese architectures have also been introduced that use two distinct branches [4]. One version of this allows the model to learn high-level semantics using contrastive losses [23].

B. *StackGAN Architecture*

For our baseline approach, we used the StackGan architecture [24]. Specifically, this is a hierarchical 2-stage GAN with two layers of GANs. The first layer outputs a lower resolution image, whereas the second layer outputs a higher resolution image and corrects any major issues. Generally, the first stage sketches primitive shapes andcolors based on the given text descriptions and the second generates high resolution, photo-realistic versions of the earlier images. **Figure** 1 shows the overall StackGAN.

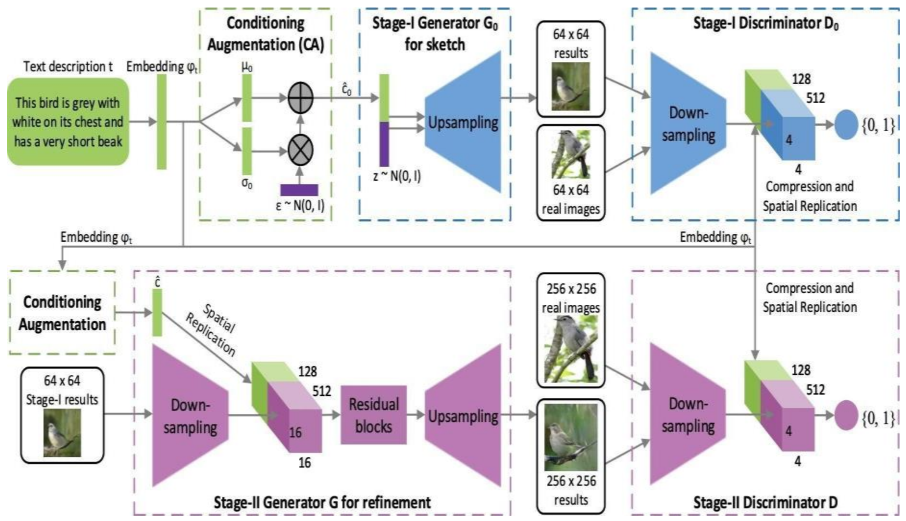

Fig. 1: The StackGAN architecture proposed in [24] that we use for our baseline and build upon.

This diagram is taken from the author's paper and gives a general overview of the training and generating process of the GAN architecture. Our implementation is adapted from the author's implementation. We retained all the components in the original architecture for our baseline and attempted improvements unless we explicitly state otherwise. Moreover, we also maintained the same hyperparameter configurations and general experimental setups unless stated otherwise.

Notable novelties of [24] include the architecture itself (which can generate high-resolution photo-realistic images) and the conditional augmentation technique to stabilize conditional GAN training and improve the diversity of induced examples. **Figure** 1 is a quite standard hierarchical GAN architecture for image generation [7]. However, the conditioning augmentation is unique. To reduce discontinuity in the latent data manifold, conditioning augmentation produces conditioning variables cˆ which are randomly sampled from some Gaussian distributionN (μ(φt), Σ(φt)) (achieved using the semi-empirical trick). Here the mean μ(φt) and diagonal covariance Σ(φt) are functions of the text embedding φt. This augmentation makes it easier for the generator to learn and introduces randomness that is beneficial for modeling text-to-image translation [24].

C. Baseline Experiments

The StackGAN model was trained on the CUB and COCO datasets and evaluated using a similar procedure to that outlined in [24] with the same meta-parameters. We verified the inception score reported for the COCO dataset and obtained an inception score of 4.956 ± 0.377 on the CUB dataset. We proceed to work on the CUB dataset for computational efficiency, so this inception score will be the baseline we hope to beat.

In the baseline there was good variety in model outputs for both the CUB and COCO datasets, indicating that the model does not suffer from a mode collapse. We note it was relatively difficult toget variety in the CUB dataset since there are more limited samples, but diversity in bird shape, color, and background was clear. Moreover, the model seemed to capture the gist of the caption well, outputting images that were largely relevant to the caption. There were two clear shortcomings of the model on the difficult MS COCO dataset. The first being the presence of artifacts in the outputs. These appear in the form of extraneous features and a lack of structure in outputs. We see that the model seems to have successfully learned and outputted relevant textures, however, it fails to assign these textures within well-defined regions/shapes. The second shortcoming is that the model isn't truly creative and likely suffers from some degree of memorization. We saw this most clearly in the case of the zebras (in COCO) where all baseline model outputs place the "zebras" in grassy fields. This is understandable since that's where we expect to find zebras rather than deserts; however, theoutputs seem to indicate that the model ignored the word "desert" entirely or somehow entangled the two concepts (zebras and grass) together from the training data (i.e. memorized). These shortcomings were also present when training the baseline on the smaller CUB dataset. We see in **Figure** 2 that the:

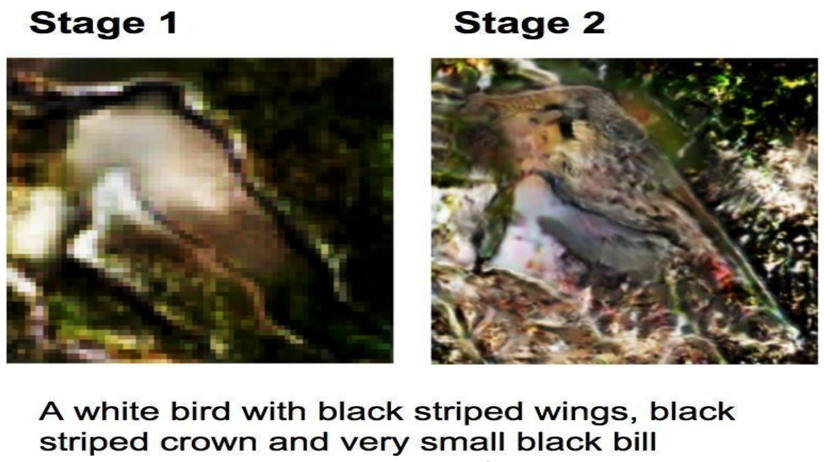
Stage 1
Stage 2
A white bird with black striped wings, black striped crown and very small black bill

Fig. 2: Note how the Stage 1 output seems to have the general structure of the image correct, but the Stage 2 output seems to lose this output structure and focuses on texture (as noted by the presence of grey feathers but the absence of anapparent head or body).

The model again struggles to assign textures to reasonable regions, often blurring any sort of bird shape to just a textured output or extending the bird region to an area that is not reasonable (i.e. the image of the white bird for our second caption in **Figure** 3). On the CUB dataset memorization is not as clear, however, the baseline model does appear to stick to the most common angle and shape of a bird in most of its outputs. Specifically, we see most baseline outputs with a recognizable bird where the bird has a short body and neck (perching-type),is rendered from a profile angle and has its wings down.

Importantly, we noticed again when training the baseline on CUB that the Stage-I output images often had better overall structure, whereas stage two had more details and better colors, often without the structure. Thus, we hypothesize that during the down-sampling or making the stitch to the Stage-II GAN,the information containing the structure is lost as shown in **Figure** 2.

II. RELATED WORK

A. Upsampling Techniques

We consider three interpolation-based methods that can be thought of as standard methods. The original model, our baseline, used nearest neighbors (NN) upsampling. This method is relatively fast, but relies on duplicating pixels and thus can lead to blocky upsampled results [20]. We also used bi-linear up-sampling, which is a more computationally intense technique that performs linear interpolation on both axes of the image, one at a time [20]. We can think of it as sampling new pixels from the line that connects the original pixels. We also considered a bi-cubic interpolation, which again increases computation and smoothing. Similar to bi-linear interpolation, cubic interpolation samples new pixels based on the cubic function that connects the previous pixels over each axis. The benefit of each of these techniques is that they require very littlememory since we are not learning any parameters for them. However, they can often lead to over-smoothed results [20].

B. Laplacian Pyramids for Up-sampling

Laplacian Pyramid architectures are very effective in image-related task super-resolution [10]. This task involves taking a low-resolution image and up-sampling to create a sharp high-resolution image. We focused on this algorithm primarily due to its lower computational cost and effectiveness compared to other learned up-sampling techniques. This architecture is unique for its feature extraction layers that occur at various points within the up-sampling pipeline[10]. To achieve this feature extraction, multiple convolutions are performed on the image. While up-sampling blocks are performed by transposed convolutions layers with learned parameters. Transposed convolutions expand the image by inserting zeros into it, then perform a convolution to get to the desired output shape. [20].

C. Residual Connections

To avoid losing the structural information from the Stage-I image, we hypothesize that explicitly carrying this information forward into the Stage-II generation stages will result in improved performance. We achieve this through the use of residual connections [5]. In addition to making deeper models easier to train, models incorporating residual connections have often achieved state-of-the-art performances in visual tasks[5] [6]. We also introduce an inception module to carry over different granularities of information in these residualconnections. [17].

D. Attention for Image Generation

Attention-based methods have become increasingly popular in recent times and many state-of-the-art approaches for any task tend to include some sort of attention mechanism in their architectures. [12] [21][18]. Since Stage-I images tend to have some relevant details for the final image but are largely blurry background noise, we hypothesize that having an attention mechanism that can attend to relevant regions of the stage-I image would provide better results.

For image generation, GANs, [25] propose a self-attention module for their architecture that leverages long-distance information during generation - overcoming a shortcoming of the largely local convolution-only GANs. Another approach, AttnGan [22] attends to the word features in different stages of the generation process. As increasingly higher resolution imagesare generated, the generation module attends to the word features so it can focus on specific details.

Implemented 5 new methods to change the up-sampling blocksor extract/propagate information through attention and residual connections. Based solely on the inception score, we see that each of the new up-sampling techniques performs better than our baseline nearest neighbors. The transposed convolutional (TC) Laplacian-based model has the highest inception score overall. We also see that both the inception-based residual model and the attention-based model with nearest neighbors perform better than our baseline. However, there is still overlap in the plausible ranges of inception scores based on the standard deviation of our inception scores.

Table 1: Inception scores of the different models along with the standard deviations ofthese scores

Model	**Inception Score** (witherror)
Baseline	4.956 ± 0.377
Bi-linear Up-sampling TC	5.198± 0.531
Laplacian Pyramid Based NN	**5.474** ± 0.505
Laplacian Pyramid Based	5.219 ± 0.453
Inception-Based Residual (with nearest neighbors)	5.031 ± 0.458
Attention-Based Model (withnearest neighbors)	5.060 ± **0.324**

Since the inception score has been criticized for being a sub-optimal metric and GANs do not have a widely accepted metric for evaluation, we also rely on qualitative analysis of our results.

From our output test images (**Figure** 3) that despite a seemingly better inception score than our baseline, bi-linear up-sampling stands out as far over-smoothed and lacking bird-like structure. This motivated us to not move forward with bi-cubic up-sampling since we expected similarly over-smooth results. In each of our other methods, we see some exceptionallylife-like birds that mostly match the captions, along with some misshapen birds. This is likely a result of the relatively small dataset size and possibly suboptimal training/model hyperparameters. We also note that despite the relatively lower inception score for our inception residual and attention-based models, they returned images comparable to the TC Laplacian Pyramid-based model. We note that the inception residual, attention, and TC Laplacian pyramid all seemingly improved on providing images that had a bird like-structure. For each of these images, it appeared much rarer to find over-smoothed andover-textured images than with our baseline, bi-linear, and the NN Laplacian model. These results:

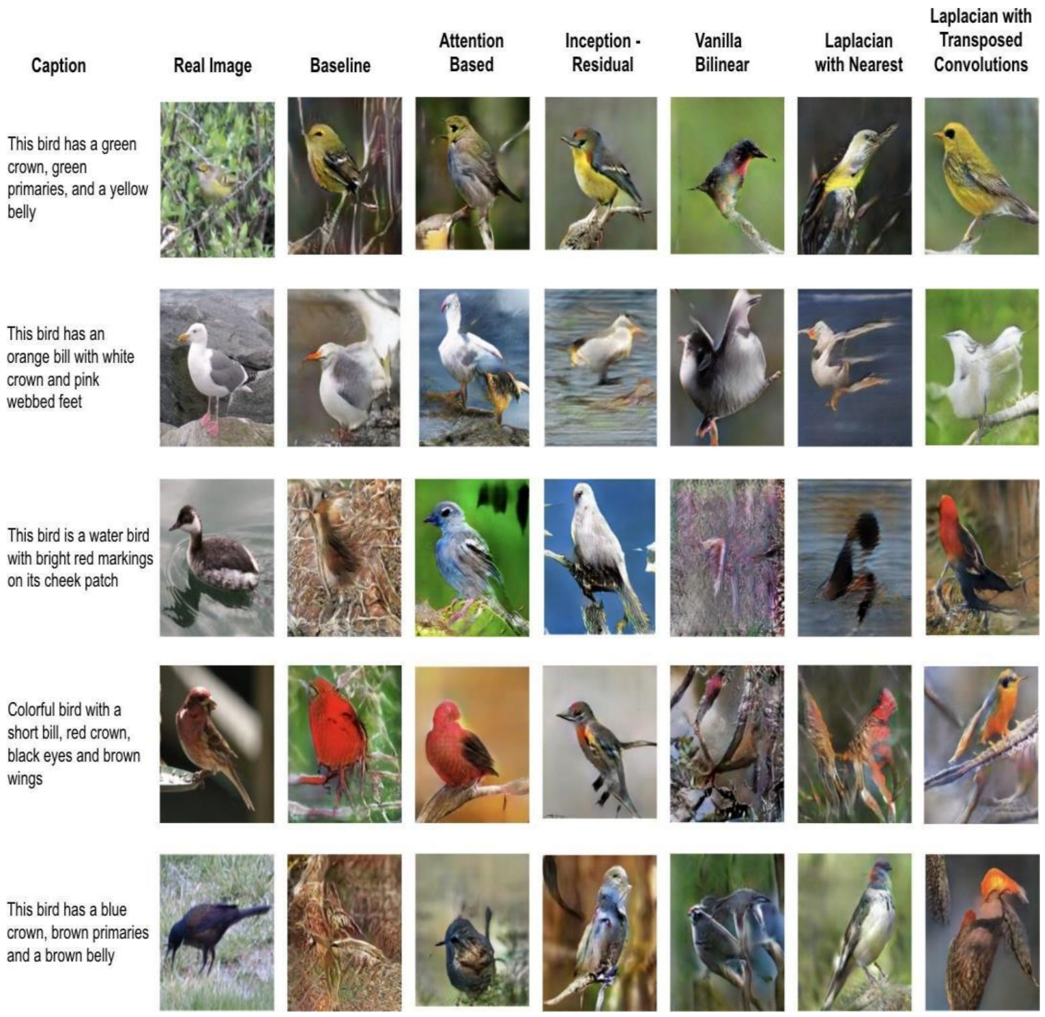
Caption
Real Image
Baseline
Attention Based
Inception - Residual
Vanilla Bilinear
Laplacian with Nearest
Laplacian with Transposed Convolutions
This bird has a green crown, green primaries, and a yellow belly
This bird has an orange bill with white crown and pink webbed feet
This bird is a water bird with bright red markings on its cheek patch
Colorful bird with a short bill, red crown, black eyes and brown wings
This bird has a blue crown, brown primaries and a brown belly

Fig. 3: Qualitative results for the different models. We see the model outputs (image columns 2-7) given a particular caption (left) and the corresponding "real"image from the dataset (image column 1).

Suggest that for these 3 models, our changes intended to increase structural information retention were successful. It also shows that the transposed convolutional layers are important for the performance of our TC Laplacian model.

To verify our models were working reasonably well we also looked for failure cases, specifically mode collapse and memorization. We more closely looked at these for the TC Laplacian pyramid-based model. We see from **Figure** 4 that this model can create distinct-looking birds based on the caption. Specifically, the final column showed 3 birds with long necks, as described in the caption. However, the first column shows 3 short-necked(perching-type) birds. We also see that thedirection of the bird changes, as well as placement and background. Though it was clear that most birds appeared perched on some sort of a branch, this is not always the case. This analysis holds for the rest of our models; however, this typeof analysis is not as relevant for our vanilla bi-linear and NN Laplacian pyramid models as they did not return reasonable birds.

III. METHODS

Our methods involved keeping the general architecture of the StackGan model and changing key components, described below, to improve picture generation. Since we found through our baseline results that Stage-I pictures appeared to hold important information around the structure of the pictures, thatwas often lost in stages when texture and color were introduced.To handle this we decided to change the upsampling blocks, as we hypothesized that the original up-sampling and downsampling blocks may blur some of the structure in the images so that it is not carried through to the Stage-II.

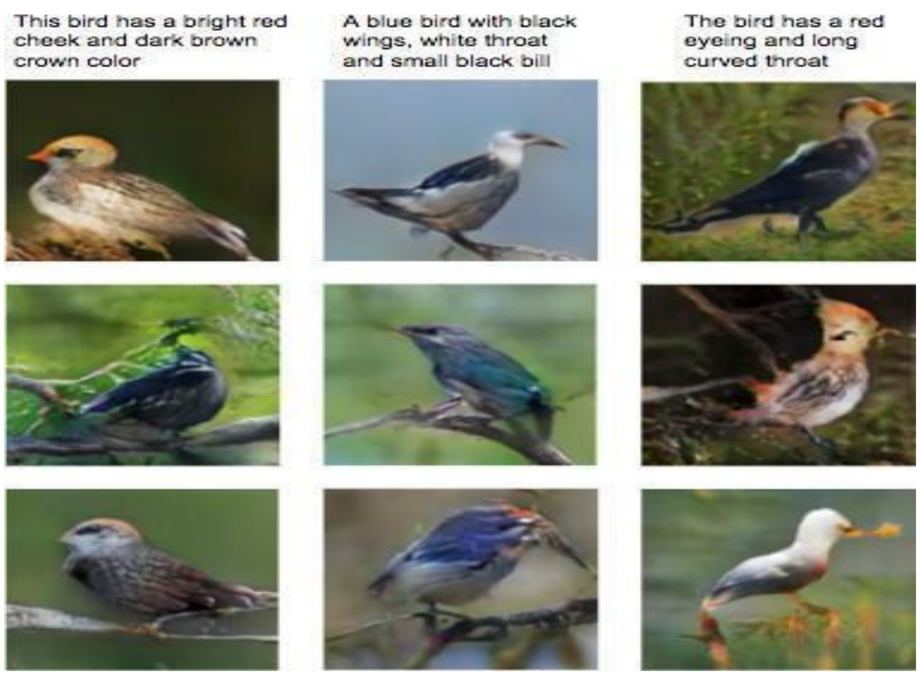
This bird has a bright red cheek and dark brown crown color
A blue bird with black wings, white throat and small black bill
The bird has a red eyeing and long curved throat

Fig. 4: Mode collapse analysis for the transposed convolution TC Laplacian pyramid-based model. We have 3 different captions, and 3 example outputs for each caption. These results indicate the absence of mode collapse in this model.

Images. Additionally, we hypothesized that adding residual or attention-based layers would also help carry over this structuralinformation.

A. *Upsampling*

Our up-sampling methods changed each up-sampling block to use a different technique in both the Stage-I and Stage-II models.

For each interpolation-based up-sampling block, nearest neighbors, and bi-linear, we performed the relevant up-sampling, followed by a 3x3 convolutional layer, a batch normalization layer, and finally a ReLU activation. For these methods, we stack multiple of these up-sampling blocks to take the image from a 192×4×4 image to 3×64×64 images. Downsampling is then performed by multiple convolutional layers. We noticed that bi-linear up-sampling over-smoothed the results and did not help to retain structural information, so we did not proceed with other interpolation-based up-sampling.

For our Laplacian pyramid-inspired architecture, an additional feature extraction block consisting of three 3x3 convolutional layers, where all but the last is followed by a ReLU activation, is used. This block has a single transposed convolution to assure the output size matches the next layer. We feed it the output of our second to last up-sampling block, then the output of this layer is added to the output of our final up-sampling block. We first used up-sampling layers that consisted of a transposed convolutions layer (TC Laplacian Based model), which matches the Laplacian Pyramid original paper. However, to investigate which part of our changes were most influential we also used nearest neighbors up-sampling followed by a 3×3 convolution (NN Laplacian Based Model).

B. *Inception-based Residual Connection*

We directly took the 64 × 64 × 3 output of the Stage-I GAN, applied a transformation to it, and added it to the inputs to the up-sampling layers in the Stage-II GAN. This transformation was an inception module [17] consisting of 1 × 1, 3 × 3, and 5 × 5convolutional layers. This would allow the Stage-II GAN to better leverage multiple granularities of structural information from Stage-I outputs, ultimately resulting in better-generated images.

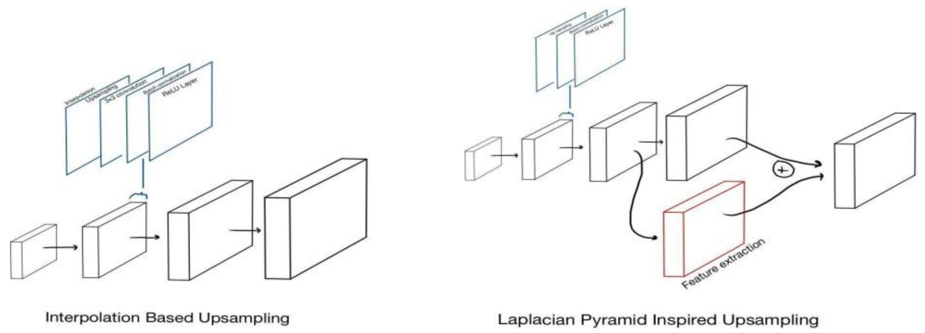

Fig. 5: Our up-sampling structure for both interpolation-based methods (left) and Laplacian Pyramid inspired architecture (right)

C. Attention-augmented Model

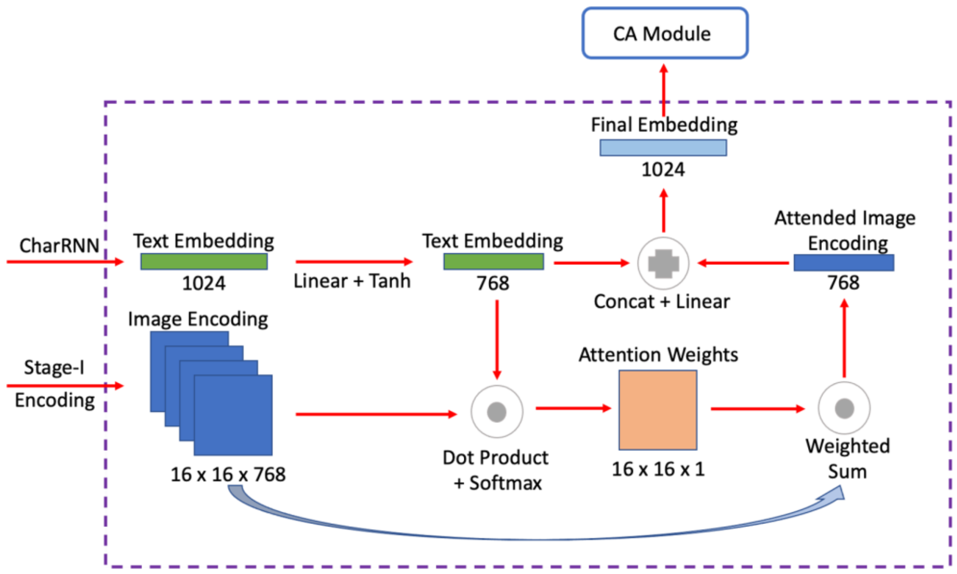

Fig. 6: Overview of the attention module implemented to better couple the visual features and text features. This module is added to the Stage-II GAN to better propagate the relevant structural information from Stage-I

For the attention module, we consider the 16 × 16 × 768 encoding of the Stage-I image and obtain a single 768 dimension vector by attending to the 1024 dimensional text embedding. This is obtained by projecting the text embedding into a 768-dimensional vector and passing it through a non-linearity. We then take the dot product of this projected text embedding with each of the 16 × 16 visual embeddings of dimensionality 768 and compute a softmax function using the values of the dot products. The softmax outputs are used as attention weights to combine the 16 × 16 × 768 dimensional encodings into a single 768-dimensional representation. This is then concatenated with the original text embedding which is finally projected to a single 1024 dimensional vector using a linear layer and then passed into the conditioning augmentationmodule of StackGAN. These steps are visually depicted in Figure 6.

This procedure allows us to more explicitly model the carrying over of caption-relevant information from the Stage-I image into the later generative layers of Stage-II, thereby minimizing the loss of important information from downsampling.

D. Evaluation

Evaluation metrics for GANs is still an area of active research. We decided to base our evaluation on the inception score. Our inception score model uses the Inception-v3 classifier pre-trained on the ImageNet dataset and returns *p*(*y x*) for eachof our generated images. Using the metric exp Ex[KL(p(y|x), p(y))] assures the generated data is similar to the data trained on (high entropy of p(y|x)) and diverse (high entropy of p(y))[16]. We note that this metric is limited in many ways, in particular since the model is trained on images other than birds,our generated images will consistently have a low probability of appearing in the data set [1].

IV. DISCUSSION

Overall, our methods appeared to improve performance on the CUB dataset by generating more realistic images to go with each caption. We notice in particular that the TC Laplacian up-sampling method appeared to improve performance compared to the baseline both in terms of inception score and quality of output images. The residual inception score and attention-based models both also generated better images, though their inception scores were not much higher than our baseline. Based on this, it is clear that explicitly passing forward information helped to minimize the loss of key information, ultimately resulting in better output images. We have shown through our work that reasonable improvements can be made tomodels simply by improving up-sampling, an area that has not been widely considered. We also note that these improvements were achieved with slightly more memory, but with very similar computational time compared to our baseline.

Thus, to boost our methods further, we could add additional feature extraction layers to our TC Laplacian pyramid-based model. We could additionally add multiple points of attentionor residual connections. Perhaps the most interesting area would be to combine the Laplacian style up-sampling with attention models that have already proved to be effective like AttnGan [22].

Additionally, a limitation in our approach was not altering the text-embeddings and relying on pre-trained char-CNN-RNN embeddings that only involved forward connections betweenwords [15]. Recent successes have been made using bi-directional recurrent neural networks that consider both forward and backward connections between words [27]. Moreover, BERT-based models might also be useful to extract more meaningful information from the text itself [3] [11]. More sophisticated techniques to learn caption representations that contain features salient to image generation would also likely result in improved performances [11]. In particular, this may help areas where our model can get the correct colors on the bird, but they are on the wrong body part (e.g. drawing red only on the crown or cheek like the 3rd and 4th caption in **Figure** 3. This is an area that would be interesting to pursue further.

We again note that there are inherent limitations to using the inception score for our quantitative evaluation metric. Inceptionscores are sensitive to the specific weights of the inceptions model, as well as the split of our test examples, two metrics that do not correspond to image generation ability [1]. Additionally, the inception score relies on the assumption that the underlyingdistribution of classes we generate matches the distribution of classes in the inception model [1]. However, since we used an inception model pre-trained on ImageNet, our classes do not line up and we may have issues due to this inherent difference between distributions [1]. An important improvement we could make would be to fine-tune or entirely train the inception modelon the CUB dataset.

REFERENCES

[1.] Shane Barratt and Rishi Sharma. A note on the inceptionscore, 2018.

[2.] Ayushman Dash, John Cristian Borges Gamboa, Sheraz Ahmed, Marcus Liwicki, and Muhammad Zeshan Afzal. TAC-GAN - Text Conditioned Auxiliary Classifier Generative Adversarial Network. *arXiv e-prints*, page arXiv:1703.06412,March 2017.

[3.] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectionaltransformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4.] Stanislav Frolov, Tobias Hinz, Federico Raue, Jörn Hees, and Andreas Dengel. Adversarial text-to-image synthesis: A review, 2021.

[5.] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[6.] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[7.] Xun Huang, Yixuan Li, Omid Poursaeed, John E. Hopcroft, and Serge J. Belongie. Stacked generative adversarial networks. *CoRR*, abs/1612.04357, 2016.

[8.] Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,39(4):664–676, 2017.

[9.] Ryan Kiros, R. Salakhutdinov, and R. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *ArXiv*, abs/1411.2539, 2014.

[10.] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *CoRR*, abs/1710.01992, 2017.

[11.] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic vision linguistic representations for vision-and-language tasks. *CoRR*, abs/1908.02265, 2019.

[12.] Elman Mansimov, Emilio Parisotto, Jimmy Ba, and R. Salakhutdinov. Generating images from captions with attention. *CoRR*, abs/1511.02793, 2016.

[13.] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[14.] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1060–1069, New York, New York, USA, 20–22 Jun 2016. PMLR.

[15.] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *CoRR*, abs/1605.05396, 2016.

[16.] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NIPS*, 2226-2234, 2019.

[17.] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[18.] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[19.] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[20.] Zhihao Wang, Jian Chen, and Steven C.H. Hoi. Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,pages 1–1, 2020.

[21.] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, AaronCourville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell Neural image caption generation with visual attention. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille,

[22.] France, 07–09 Jul 2015. PML

[23.] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *CoRR*, abs/1711.10485, 2017.

[24.] Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. Semantics disentangling for text-to-image generation, 2019.

[25.] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and

[26.] Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *2017 IEEE International Conference on*

Computer Vision (ICCV), pages 5908–5916, 2017.

[27.] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019.

[28.] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *CoRR*, abs/1710.10916, 2017.

[29.] Tehseen Zia, Shahan Arif, Shakeeb Murtaza, and Mirza Ahsan Ullah. Text-to-image generation with attention-based recurrent neural networks, 202