Multi-Class Multi-Instance Count Conditioned Adversarial Image Generation -Appendix-

Amrutha Saseendran[1], Kathrin Skubch[1] and Margret Keuper[2]
[1]Bosch Center for Artificial Intelligence, [2]University of Siegen
Amrutha.Saseendran@de.bosch.com

In this document, we provide additional details and results to the main paper. The document is structured as follows:

1. Additional quantitative analysis

1.1. CLEVR

The quantitative evaluation for CLEVR simple images where similar shapes are constrained to same color (red cylinders, green spheres and blue cubes) are shown in Table 1.

Dataset	Acc(↑)	FID(↓)
CLEVR-2	0.99	6.524
CLEVR-3	0.98	8.023

Table 1: Average count accuracy (Acc) and Fréchet Inception Distance (FID) - CLEVR simple dataset.

2. Extended analysis across dataset

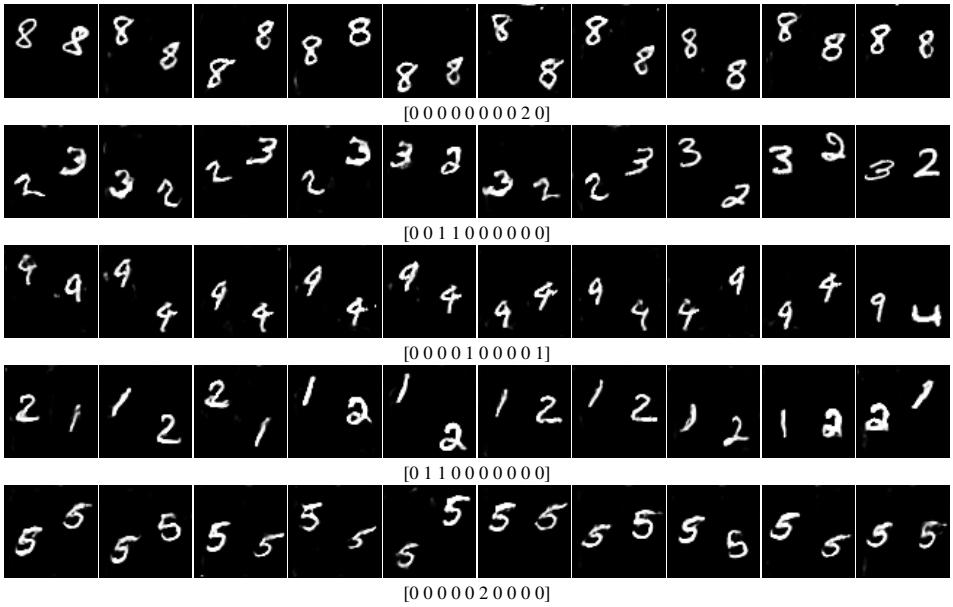
2.1. Multi MNIST

We consider Multi-MNIST dataset to evaluate the general capability of our approach to generate images based on learned count information using MC^2-SimpleGAN. We evaluate on two variants of the Multi-MNIST dataset, one with two digits per image (MNIST-2) and the other with three digits per image (MNIST-3). The datasets were generated by uniformly sampling digits from the MNIST dataset and placing them in non-overlapping positions on black background. We used 1000 images for each digit combination during training. The count information is provided to the model as a vector with ten entries comprising the desired number of instances of each digit in the image. Figure 1 shows the generated samples by our model for different count combinations. The results show that the proposed model is able to produce images based on the given digit count without any supervision. We observed an average count accuracy of 96% for MNIST-2 images and 93% for MNIST-3, where the accuracy slightly decreases for higher counts.

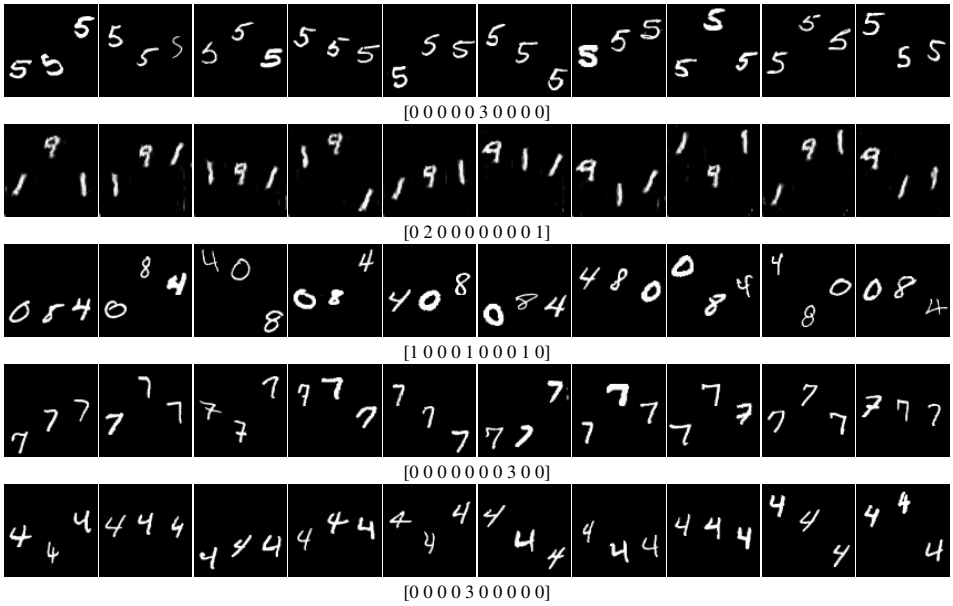
Interpolation and Extrapolation To check for interpolation and extrapolation ability of the model, we trained the network with images containing only certain combinations of input count and during testing we input an unseen count combination of digits (see Figure 1c). For the MNIST-2 dataset, the model sees the count value of only 1 for digit 6 during training. Similarly for the MNIST-3 dataset, the count 3 for digit 9 is unseen during training. The model is able to transfer the concept of count from one digit to another and predict the count for these unseen combinations. For count value 0 for all digits (also unseen during training), the model is able to generate images without any digits.

2.2. Extended quantitative analysis across datasets

The multiple count prediction distribution of cylinder and sphere class of CLEVR-2 and that of cylinder, sphere



(a) Generated MNIST-2 images.



(b) Generated MNIST-3 images.

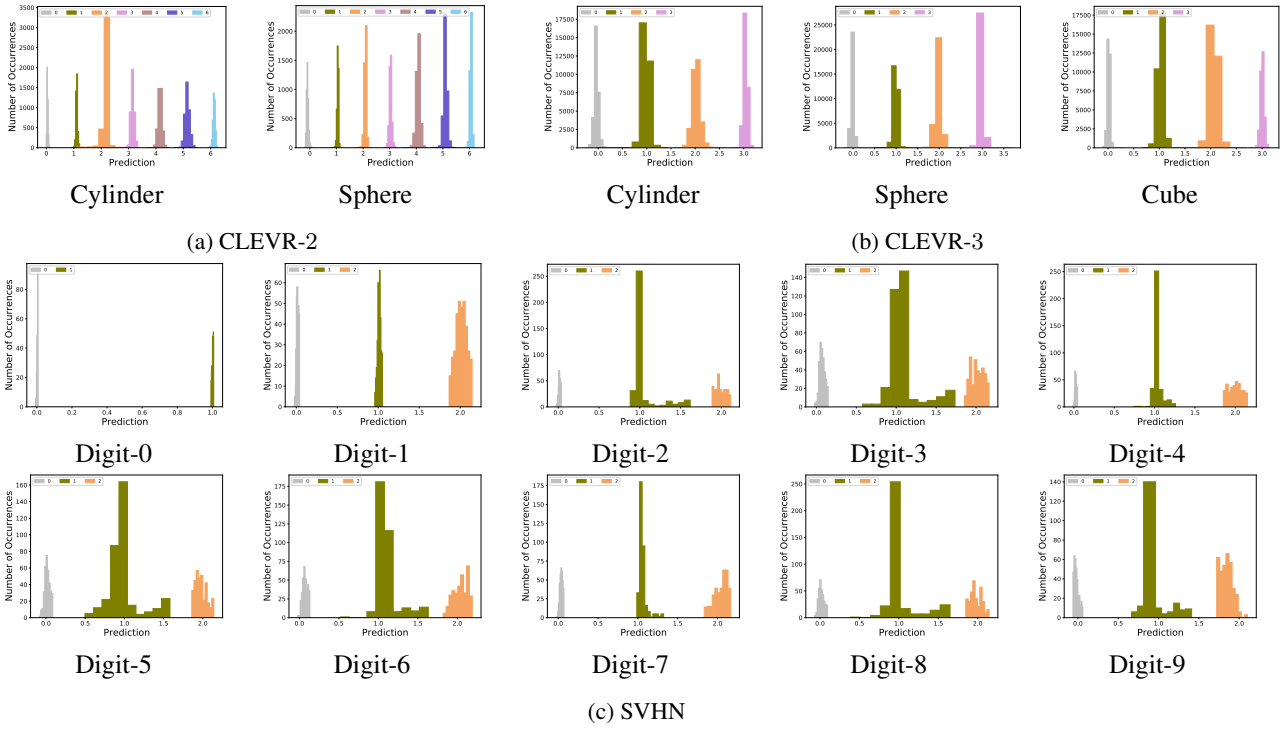
(c) Generated MNIST images for unseen count combination.

Figure 1: Generated Multi-MNIST images for different ten-dimensional count vectors - MC^2-SimpleGAN.

and cube class of CLEVR-3 are shown in Figure 2a and 2b respectively. Similarly the multiple count distribution for the ten digit classes in SVHN are visualized in Figure 2c.

2.3. Extended qualitative analysis across datasets

We provide additional qualitative results for CLEVR, SVHN and CityCount images for various count combina-



(a) CLEVR-2

(b) CLEVR-3

(c) SVHN

Figure 2: Count performance on CLEVR and SVHN images. The figure shows the predicted count values for each count class.

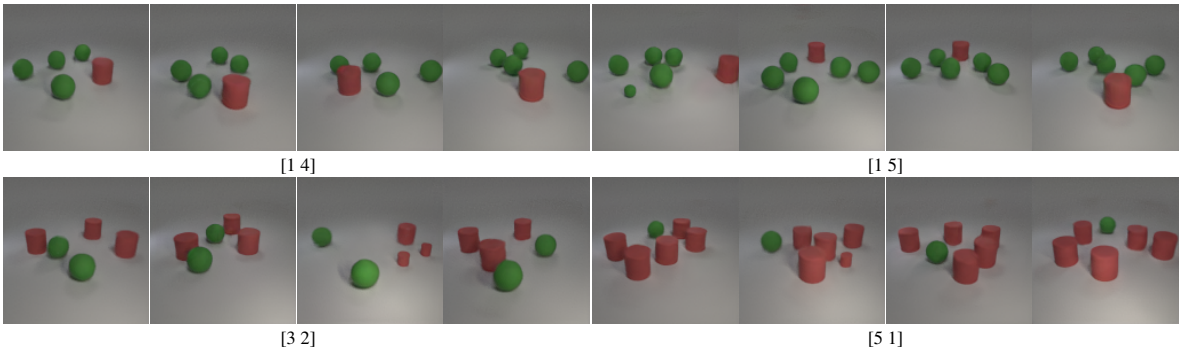
tions. The generated images for CLEVR-2 and CLEVR-3, both with same color and multicolor for similar shapes are shown in Figure 3. The additional results for SVHN images along with their respective input count are shown in Figure 4. The extended results for CityCount images along with their respective input count are shown in Figure 5. For the ease of visualization boxes are drawn around objects of interest. The generated images exhibit diversity and good quality (FID values are given in the main paper) across all the datasets considered.

3. Network architecture and implementation details

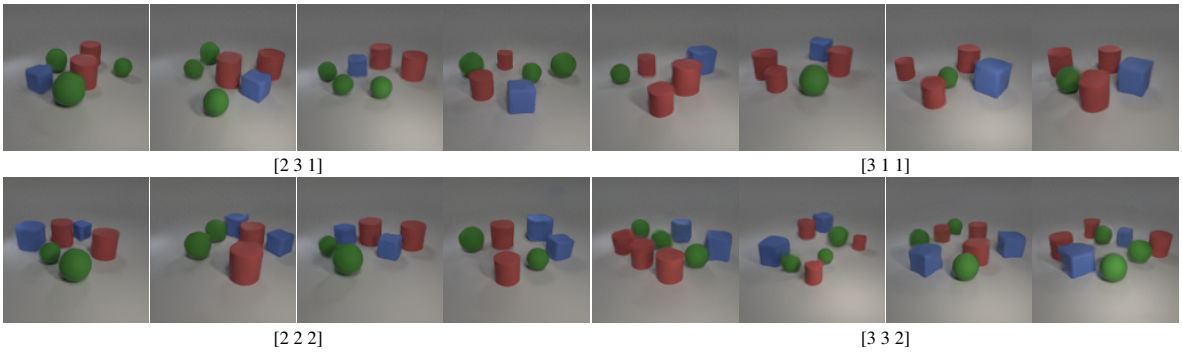
3.1. MC^2-SimpleGAN

We introduce MC^2-SimpleGAN as a simple network based architecture of our proposed method for fair and easy comparison study with other conditional GAN variants. The generator of the MC^2-SimpleGAN is shown in Figure 6. The architecture comprises of a count conditioned generator and a discriminator , that is equipped with an additional count prediction network. Our basic generator network is inspired from the DenseNet [2] architecture. DenseNet introduces dense blocks which consist of several convolutional layers where the output from each layer is connected in a feed forward fashion to its succeeding layers (see Figure 6a for a visualization). The additional skip connections in the dense blocks strengthen the count conditioning in the generator since the input feature maps are connected to the output layers of the dense block [1]. We use two dense blocks of three layers with a growth rate of 64. The generator (Figure 6b) gets as input a combination of randomly sampled noise and a multiple class count vector. The concatenated vectors are passed through a fully connected layer (FC) with ReLU activation which is followed by the dense blocks. The two dense blocks are coupled with a 1×1 convolution to decrease the number of output feature maps and to improve computational efficiency [2] and an upsampling layer to increase the spatial resolution. The output feature maps from the dense block layers are forwarded to two 3×3 convolutional layers (Conv) to generate images. For the discriminator as well as for the count network, we use four convolutional layers with shared weights followed by a fully connected layer to discriminate between real and fake images (discriminator) or to regress the multiple class count vector (count network).

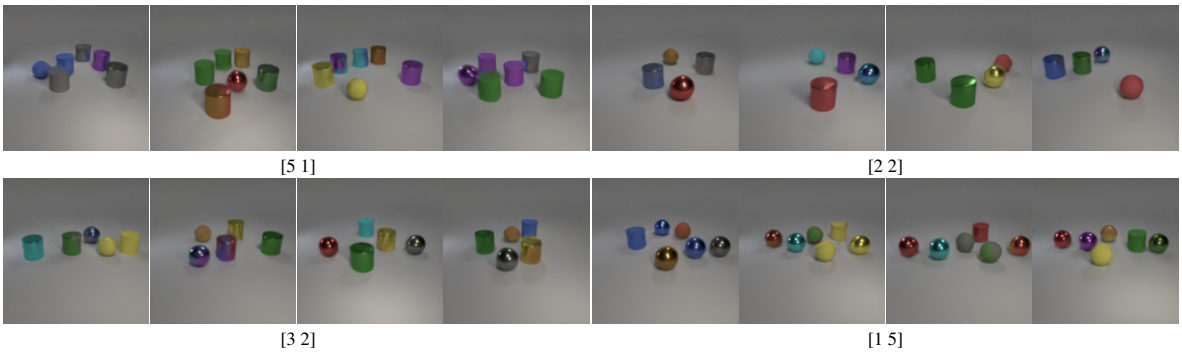
Implementation Details The models are trained with images of size 64×64 for Multi-MNIST and SVHN and 128×128 for CLEVR images. All images were scaled at the input with pixel values ranging from -1 to 1. Adam



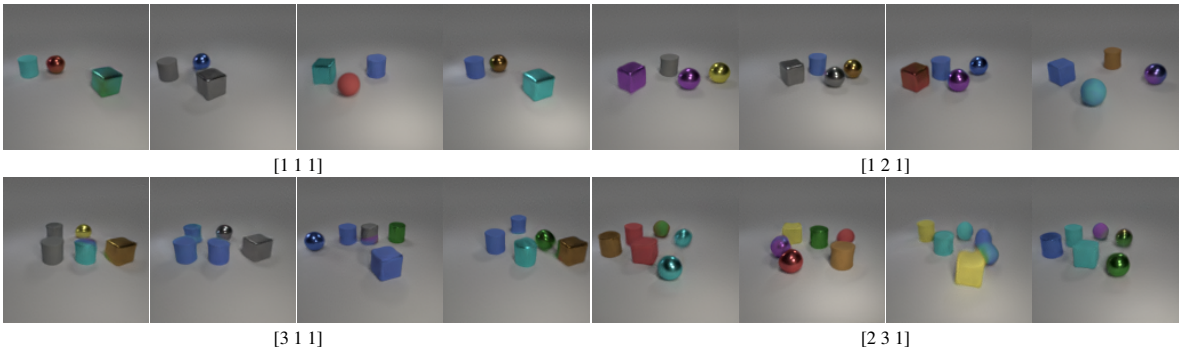
(a) CLEVR-2 .



(b) CLEVR-3.



(c) CLEVR-2 multicolor.



(d) CLEVR-3 multicolor.

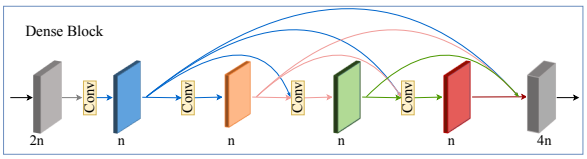
Figure 3: Generated MC^2-StyleGAN2 CLEVR images for different count combinations. The result shows better quality images with correct count condition.

(a) Generated SVHN-2 images.

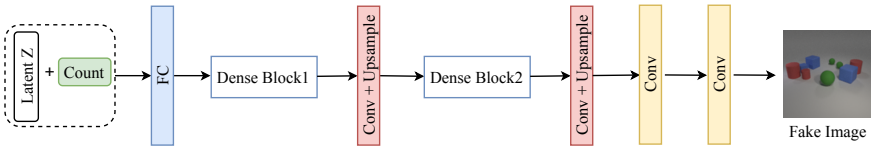
Figure 4: Generated MC^2-StyleGAN2 SVHN-2 images for different ten-dimensional count vectors. The generated images are of better image quality with correct number of object instances according to the condition.

(a) Real and Generated CityCount images.

Figure 5: Real and generated CityCount images by our MC^2-StyleGAN2. The model generates good quality images with correct number of cars/persons.



(a) Dense block: Three layer dense block with growth rate n



(b) MC^2-SimpleGAN Generator: A noise sample and a multi-class count vector are passed to a fully connected layer (FC). Two dense blocks (details see above) coupled with conv. and upsampling layers are followed by two conv. layers.

Figure 6: MC^2-SimpleGAN Generator.

optimizer is used with momentum weights, $\beta_1 = 0.5$ and $\beta_2 = 0.999$ respectively. For generator and discriminator learning rate is fixed to $1e-4$. The network is trained for 200 epochs with batch size 128 and count loss co-efficient λ as 0.7.

3.2. MC^2-StyleGAN2

We extended the official StyleGAN2 TensorFlow implementation [4] corresponding to configuration-e for our count based image generation for CLEVR and SVHN images. Since the number of training images for CityCount is low, adaptive discriminator augmentation [3] was applied while training the networks for CityCount images. The mapping network is concatenated with the multiple-class count vector at each layers. We also introduce dense like connections in place of residual connections in the synthesis/generator network for improved results. We calculate the FID values on five samples of 50k generated images and report the average value.

Implementation Details

CLEVR and SVHN The models are trained with images of size 64×64 for SVHN and 128×128 for CLEVR images. The network is trained with minibatch size 32 and the count loss co-efficient as 0.7. The rest of the hyperparameters is similar to the official implementation [4].

CityCount The models are trained with images of size 256×256 for CityCount images. For generator and discriminator, the learning rate is fixed to 0.0025, and Adam optimizer is used with momentum weights, $\beta_1 = 0.0$ and $\beta_2 = 0.99$ respectively. The number of mapping layers used in the mapping network is 4. The rest of the hyperparameters is similar to the official implementation [3].

3.3. Count prediction network

A convolution based network architecture shown in Figure 7 is used for the prediction of multiple-class count prediction of images. The count regression network is similar to the one used in the count sub-network of the MC^2-SimpleGAN. The network includes four convolution layers followed by LeakyRelu activation and dropout layers with the final block as a fully connected layer which outputs the multiple-class count vector of the corresponding input image.

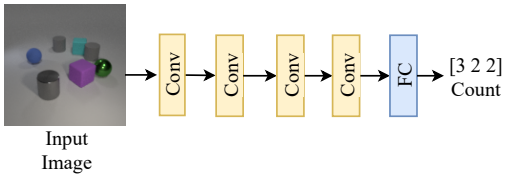


Figure 7: Count prediction network for CLEVR images. The network predicts the number of cylinders, spheres and cubes in the input RGB images as count vector

References

[1] Yuhua Chen, Feng Shi, Anthony G. Christodoulou, Zhengwei Zhou, Yibin Xie, and Debiao Li. Efficient and accurate mri super-resolution using a generative adversarial network and 3d multi-level densely connected network. In *MICCAI*, 2018. 3

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. 3

[3] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020. 7

[4] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 7