

# Mini-batch Similarity Graphs for Robust Image Classification - Supplementary material

Arnab Mondal\*

<https://mila.quebec/en/person/arnab-mondal/>

Vineet Jain\*

<https://mila.quebec/en/person/vineet-jain/>

Kaleem Siddiqi

<http://www.cim.mcgill.ca/~siddiqi/>

Centre for Intelligent Machines

School of Computer Science

McGill University

Montréal, Canada

This is supplementary material primarily contains:

- Additional classification results
- More ablation studies
- Robustness results for all model variations
- Connection to mini-batch discrimination

## 1 Additional Classification results

### 1.1 Training plots

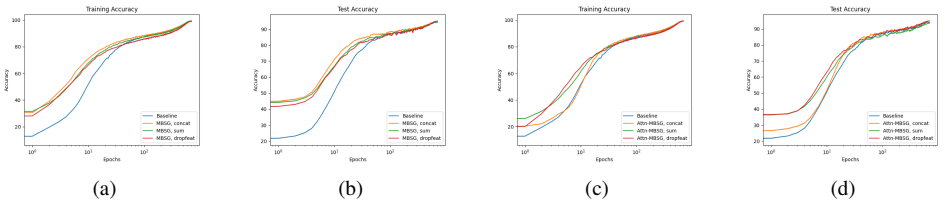


Figure 1: We encourage the reader to zoom-in on the PDF. The plots show training and test accuracy versus epochs, for different models on the CIFAR-10 dataset using ResNet-50 encoder. The epochs axis is on a log scale. (a) training accuracy for MBSGs, (b) test accuracy for MBSGs, (c) training accuracy for attention MBSGs, (d) test accuracy for attention MBSGs.

Figure 1 compares the training and test accuracy of all the models, with the standard supervised baseline versus the number of epochs on the CIFAR-10 dataset. The MBSG models train faster than the standard network, giving a significant performance difference during early parts of the training process.

## 1.2 More ablation studies

In this section, we provide classification results while changing certain hyperparameters of our model. We perform all these experiments on CIFAR-10 dataset using a ResNet-50 base encoder and a single layer MBSG with batch size 256 and neighborhood size  $k = 16$ .

**Attention heads:** Different number of attention heads (N) for the Attn-MBSG model. Table 1 shows that increasing number of attention heads up to three improves performance, after which it plateaus. Our experiments show that  $N=3$  leads to optimal performance.

Model	N=1	N=2	N=3	N=4	N=5
Attn-MBSG (dropfeat)	94.60	94.89	95.05	95.03	95.04

Table 1: Image classification results on CIFAR-10 using a Resnet-50 encoder and an Attn-MBSG, with different number of attention heads N.

**Weighted Addition:** Different values for the weighted addition parameter  $\beta$  for combining self and neighbor information. Table 2 shows that as  $\beta$  increases, the performance increases until it reaches an optimal value, and then it starts decreasing. The optimal value from our experiments was found to be  $\beta = 0.5$ .

Model	$\beta=0.0$	$\beta=0.25$	$\beta=0.50$	$\beta=0.75$	$\beta=1.0$
MBSG (sum)	85.45	93.42	95.02	94.83	94.44
Attn-MBSG (sum)	86.10	93.82	94.95	94.91	94.30

Table 2: Image classification results on CIFAR-10 using a Resnet-50 encoder and an MBSG using weighted addition, with different values for  $\beta$ .

**Drop Feature:** Different values for the drop feature probability  $p$  for selecting either self or neighbor information. Table 3 shows that as  $p$  increases, the performance increases until it reaches an optimal value, and then it starts decreasing. The optimal value from our experiments was found to be  $p = 0.5$ .

Model	$p=0$	$p=0.25$	$p=0.50$	$p=0.75$	$p=1$
MBSG (dropfeat)	85.40	93.90	95.24	94.90	94.44
Attn-MBSG (dropfeat)	86.10	94.10	95.05	94.80	94.30

Table 3: Image classification results on CIFAR-10 using a Resnet-50 encoder and an MBSG using drop feature, with different values for  $p$ .

### 1.3 Results for WideResnet

Table 4 provides results for our MBSG with all the combine options, using Wide ResNet-28-10 as the encoder network. We also use this network for all the baselines in this table. We observe a consistent improvement across all datasets, which is consistent with the results of our experiments using ResNet-50 in the main paper.

Model	CIFAR 10		CIFAR 100		MIT 67	
	Inductive	Transductive	Inductive	Transductive	Inductive	Transductive
Supervised vanilla	95.62 $\pm$ 0.14		79.58 $\pm$ 0.20		66.20 $\pm$ 0.19	
Supervised contrastive [9]	95.91 $\pm$ 0.16		80.15 $\pm$ 0.15		66.89 $\pm$ 0.17	
Affinity supervision [14]	95.59 $\pm$ 0.18		79.8 $\pm$ 0.19		66.8 $\pm$ 0.16	
MBSG (concat)	96.02 $\pm$ 0.21	96.05 $\pm$ 0.20	80.18 $\pm$ 0.18	80.20 $\pm$ 0.18	66.87 $\pm$ 0.22	66.90 $\pm$ 0.21
MBSG (sum)	95.78 $\pm$ 0.15	95.80 $\pm$ 0.14	79.85 $\pm$ 0.17	79.88 $\pm$ 0.15	66.52 $\pm$ 0.18	66.51 $\pm$ 0.17
MBSG (dropfeat)	<b>96.14</b> $\pm$ 0.16	<b>96.17</b> $\pm$ 0.18	80.46 $\pm$ 0.21	80.45 $\pm$ 0.20	67.10 $\pm$ 0.19	67.12 $\pm$ 0.20
Attn-MBSG (concat)	95.95 $\pm$ 0.15	95.95 $\pm$ 0.18	80.22 $\pm$ 0.20	80.23 $\pm$ 0.19	66.96 $\pm$ 0.18	66.98 $\pm$ 0.18
Attn-MBSG (sum)	95.89 $\pm$ 0.19	95.91 $\pm$ 0.18	80.02 $\pm$ 0.20	80.06 $\pm$ 0.18	66.67 $\pm$ 0.16	66.69 $\pm$ 0.17
Attn-MBSG (dropfeat)	96.12 $\pm$ 0.20	96.14 $\pm$ 0.21	<b>80.76</b> $\pm$ 0.17	<b>80.76</b> $\pm$ 0.15	<b>67.25</b> $\pm$ 0.22	<b>67.26</b> $\pm$ 0.20

Table 4: Image classification results using a Wide ResNet-28-10 encoder. The architectures are trained using a batch size of 256 and with  $k = 16$  for CIFAR-10, and  $k = 4$  for CIFAR-100 and MIT67. We provide results for different combine modes of our single layered mini-batch graph based models (rows 4-9).

## 2 Additional robustness results

### 2.1 Random Perturbations

We provide results for Gaussian noise and Gaussian blurring perturbation for all the different variations of the MBSG models, using the ResNet-50 encoder on the CIFAR-10 dataset. Figure 2 and 4 show some sample CIFAR-10 images, with different levels of corruption severity for visualization. For both Gaussian noise and Gaussian blurring we define corruption severity as the standard deviation,  $\sigma$  used when sampling from the Gaussian distribution, with higher values of  $\sigma$  corresponding to increased corruption.

Figure 3 shows the average test accuracy for different levels of Gaussian noise and Figure 5 shows the average test accuracy for different levels of Gaussian blurring. In both figures, the top row shows plots for MBSG models (concat, sum and dropfeat) and the bottom row shows plots for Attn-MBSG models (concat, sum and dropfeat). Models using MBSGs (purple) maintain higher accuracy over the entire range of corruption severities as compared to the baseline model (blue) and show a lower drop in accuracy for higher corruption levels. We also observe that the sum combination method generally performs better than the other variations.

### 2.2 Effect of neighbourhood size on robustness

We study the effect of the neighbourhood size on the robustness of the model we look at the our MBSG (sum) models performance when fix a corruption severity for both Gaussian noise and Gaussian blurring and change the number of neighbours  $k$ .

Corruption type (Corruption Severity)	$k=0$	$k=4$	$k=8$	$k=16$	$k=32$
Gaussian Blurring (Blur intensity = 1)	89.60	90.20	90.61	91.63	91.82
Gaussian Noise (Standard deviation = 0.1)	19.10	22.32	26.81	34.82	36.27

Table 5: Image classification results for corrupted images on CIFAR-10 using a Resnet-50 encoder and an MBSG (sum), with different values for  $k$ .

Table 5 shows that the performance is indeed directly proportional  $k$ . Hence, the error should be inversely proportional to  $k$  as claimed in Proposition 1.

### 2.3 Black-box Adversarial Attacks

We provide histogram plots of the number of queries required until a successful attack (over 1000 images) for all the variations of the MBSG model. Figure 6 and 7 show the plots for SimBA [9] and Bandits-TD [9], respectively. The dashed lines indicate the median value and the dotted lines indicate the mean value for the different models. In both figures, the top row shows plots for MBSG models (concat, sum and dropfeat) and the bottom row shows plots for Attn-MBSG models (concat, sum and dropfeat). The performance across the different combination methods is similar, although dropfeat models generally have a higher number of mean and median queries, due to the heavier tail in the distribution. Also, the MBSG models with attention mechanism have a lower number of mean and median queries on average than the models without attention.

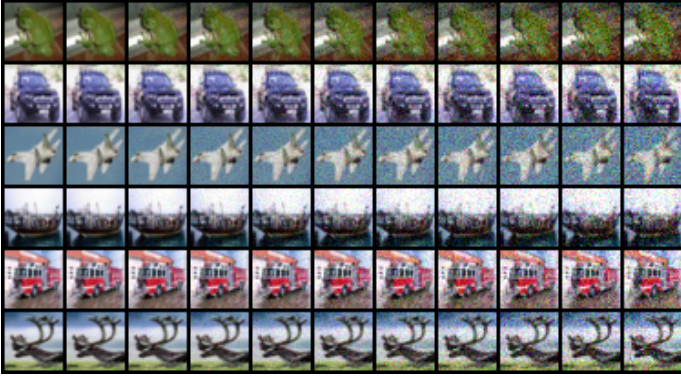
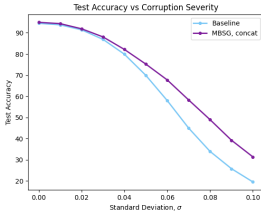
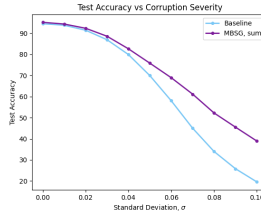


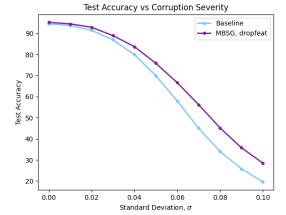
Figure 2: Sample images from the CIFAR-10 dataset with each column showing an increasing level of corruption severity, for pixelwise Gaussian noise.



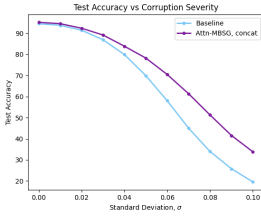
(a) MBSG, concat



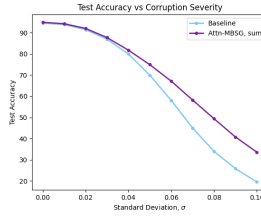
(b) MBSG, sum



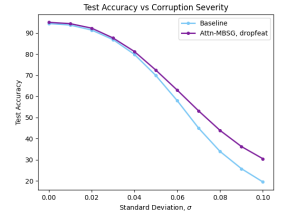
(c) MBSG, dropout



(d) Attn-MBSG, concat



(e) Attn-MBSG, sum

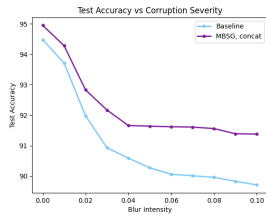


(f) Attn-MBSG, dropout

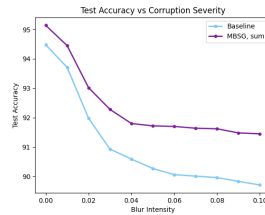
Figure 3: Average test accuracy at different corruption severities for Gaussian noise on CIFAR10, using ResNet-50 with MBSGs (top) and ResNet-50 with Attention MBSGs (bottom). Models using MBSGs (purple) maintain higher accuracy over the entire range of corruption severities as compared to the baseline model (blue), and show a lower drop in accuracy for higher corruption levels.



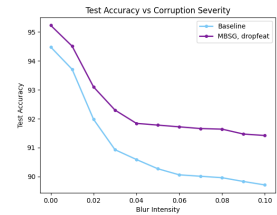
Figure 4: Sample images from CIFAR-10 dataset with each column showing increasing level of corruption severity for Gaussian blurring.



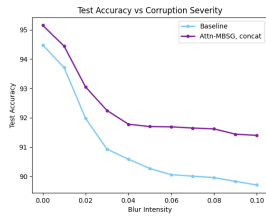
(a) MBSG, concat



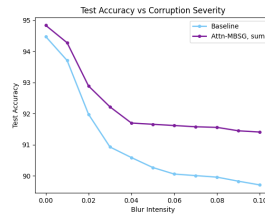
(b) MBSG, sum



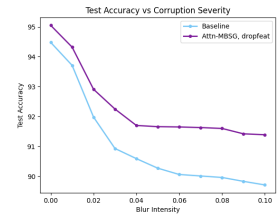
(c) MBSG, dropout



(d) Attn-MBSG, concat



(e) Attn-MBSG, sum



(f) Attn-MBSG, dropout

Figure 5: Average test accuracy at different corruption severities for Gaussian blurring on CIFAR10, using ResNet-50 with MBSGs (top) and ResNet-50 with Attention MBSGs (bottom). Models using MBSGs (purple) maintain higher accuracy over the range of corruption severities as compared to baseline model (blue) and have lower drop in accuracy for higher corruption levels.

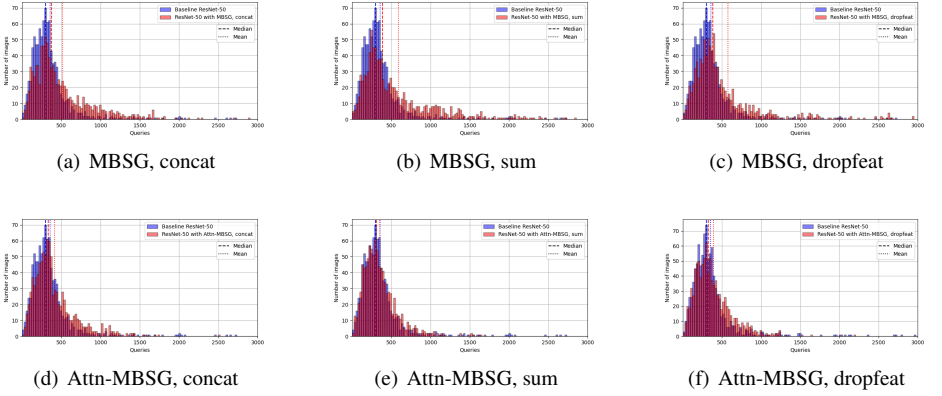


Figure 6: Histogram of number of queries required until a successful attack (over 1000 target images) using simBA on the CIFAR10 dataset. The queries axis is limited to 3000 queries for clarity of presentation. Models using MBSGs (red) require a larger number of queries on average for a successful attack as compared to the baseline model (blue).

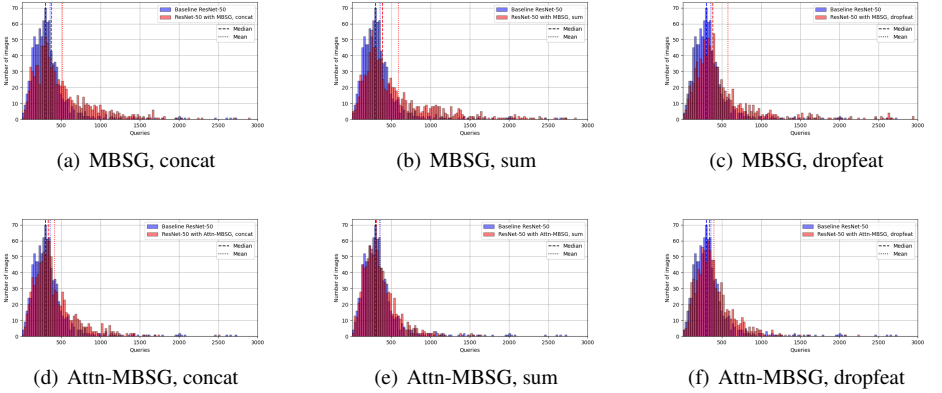


Figure 7: Histogram of number of queries required until a successful attack (over 1000 target images) using Bandits-TD on the CIFAR10 dataset. The queries axis is limited to 3000 queries for clarity of presentation. Models using MBSGs (red) require a larger number of queries on average for a successful attack, as compared to the baseline model (blue).

### 3 Connection to Mini-batch Discrimination

Generative Adversarial Networks (GANs), first introduced in [2], are a family of generative models that are used in several computer vision tasks including high resolution image generation[1], image super-resolution[2], domain adaptation[5, 4] and image compression [3]. GANs suffer from the problem of mode collapse, where the generated samples belong to a few modes in the dataset while still being successful at fooling the discriminator. This leads to a lack of diversity in the generated samples. One way to mitigate mode collapse in GANs is to use a technique known as *mini-batch discrimination* [2]. Here, instead of the discriminator being required to label individual samples as fake or real, it discriminates between an entire mini-batch of generated or real samples.

As it turns out, our proposed Attn-MBSG can be interpreted as an extension of mini-batch discrimination to the classification task. To establish this connection, we present a variation of our MBSG, which we refer to as a Mode Collapse MBSG (MC-MBSG). Rather than aggregating node features weighted by attention, as in the case of Attn-MBSG layers, we aggregate the edge features in this model. Note that these edge features capture similarity and can be interpreted as the unnormalized attention values.

In MC-MBSG, we modify the discriminator network by extracting features from real/fake images using the encoder,  $f_\theta(\cdot)$ , and denote them by  $\{h_1, h_2, \dots, h_B\}$  where  $h_i = f_\theta(x_i)$  and  $B$  is the batch size. We induce complete graphs for both the batch of generated and real samples, and process them separately. The output for the  $n$ -th aggregated edge feature for the  $i$ -th sample in the mini-batch, which is computed in a manner similar to aggregating unnormalized attention weights for the  $n$ -th head in Attn-MBSG, is given by:

$$\bar{h}_i^n = \sum_{j=1}^B \phi(\psi(W_n h_i, W_n h_j)), \quad (1)$$

where  $\phi(\cdot)$  is a neural network,  $W_n$  is a trainable matrix, and  $\psi$  can either be concatenation or absolute difference. We can then concatenate the aggregated edge features with the independent node features and use a final layer to get the  $i$ -th scalar output:  $o_i = \sigma(W_{final}(h_i \parallel \bar{h}_i^1 \parallel \dots \parallel \bar{h}_i^N))$ , where  $\sigma(\cdot)$  is the sigmoid function. If we restrict  $\psi(\cdot)$  to absolute difference,  $\phi(\cdot)$  to a fixed  $\exp(-x)$  function and take all the  $\bar{h}_i^n$  as scalars then the model reduces to standard mini-batch discrimination [2]. This shows how MBSGs are connected to mini-batch discrimination. Our MC-MBSG model is more expressive than mini-batch discrimination, and can significantly mitigate mode collapse in GANs, as demonstrated by the experiments in Section 3.1.

#### 3.1 GAN training using MC-MBSG

Lastly, we provide results for training GANs using MC-MBSG, and compare this strategy with both a vanilla GAN and minibatch discrimination [2]. We test the diversity of our generated samples using the technique of Number of statistically Different Bins (NDB), which was proposed in [8] as a metric to quantify mode collapse in GANs. To compute this metric, we first cluster the training dataset in  $K$  different bins and then allocate the generated images in these bins based on their proximity to the centroid of each bin. Then, we measure the statistical similarity between the real and fake images in each of the bins and compute the fraction of statistically different bins that give us the NDB score. In the case of mode collapse, the number of statically different bins is close to  $K$ , and the NDB score is close to 1. Using the NDB score as the metric, in Figure 8, we show that our proposed MC-MBSG



helps the generator learn faster and generate more diverse samples, which is substantiated by lower NDB scores. We provide the details of the architecture, experiment setup and generated images in the Supplementary material.

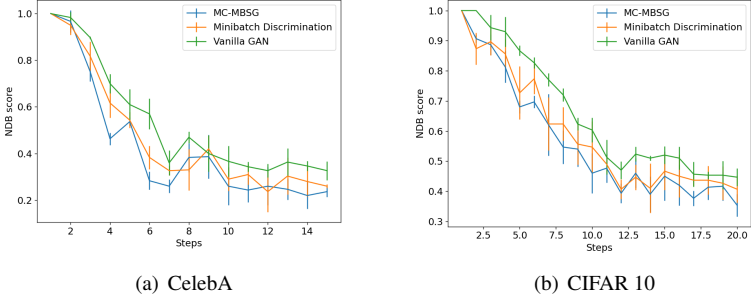


Figure 8: Plots of NDB scores for 100 bins over the steps, where a step refers to 500 training iterations. Lower NDB scores imply higher sample diversity. We provide the confidence of the NDB values over 5 runs. We take 160 batches of real training images and 40 batches of fake generated images with a batch size of 128 to estimate the true statistics of the dataset and reduce the NDB test time.

### 3.2 Details of the GAN architecture

We use standard generator and discriminator architectures for our GAN model. Let us denote the following operations,

Basic convolution:  $\text{Conv}(\text{in\_channels}, \text{out\_channels}, \text{filter\_size}, \text{stride}, \text{padding})$ ,

Linear layer:  $\text{Linear}(\text{in\_dim}, \text{out\_dim})$ ,

Deconvolution:  $\text{Conv\_trans}(\text{in\_channels}, \text{out\_channels}, \text{filter\_size}, \text{stride}, \text{padding}, \text{output\_padding})$

The generator architecture using the above notation, is given by

$\text{Linear}(128, 16384) - \text{batch\_norm} - \text{ReLU}$

$\text{Conv\_trans}(1024, 512, 5, 2, 2, 1) - \text{batch\_norm} - \text{ReLU}$

$\text{Conv\_trans}(512, 256, 5, 2, 2, 1) - \text{batch\_norm} - \text{ReLU}$

$\text{Conv\_trans}(256, 128, 5, 2, 2, 1) - \text{batch\_norm} - \text{ReLU}$

$\text{Conv\_trans}(128, 64, 5, 2, 2, 1) - \text{batch\_norm} - \text{ReLU}$

$\text{Conv\_trans}(64, 3, 5, 1, 2, 0) - \text{tanh}$

The discriminator architecture is given by

$\text{Conv}(3, 64, 4, 2, 1) - \text{batch\_norm} - \text{LeakyReLU}$

$\text{Conv}(64, 128, 4, 2, 1) - \text{batch\_norm} - \text{LeakyReLU}$

$\text{Conv}(128, 256, 4, 2, 1) - \text{batch\_norm} - \text{LeakyReLU}$

$\text{Conv}(256, 512, 4, 2, 1) - \text{batch\_norm} - \text{LeakyReLU}$

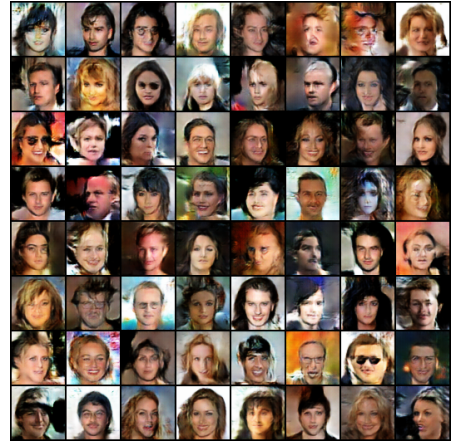
$\text{Linear}(8192, 1) - \text{sigmoid}$

This defines the GAN architecture for the CelebA dataset. For CIFAR-10, we slightly modify the  $\text{Linear}(\cdot)$  layers to adjust to the reduced image sizes. We add both the minibatch discrimination and MC-MBSG layers before the final  $\text{Linear}(\cdot)$  layer of the discriminator. For the sake of our experiments, we use the Adam optimizer with a learning rate of  $10^{-4}$  for the discriminator and  $2 \times 10^{-4}$  for the generator, with a batch size of 128. We provide the

generated images using different discriminator architectures in Figure 9 and 10. We train the model on CIFAR-10 for 10000 iterations and on CelebA for 7500 iterations.

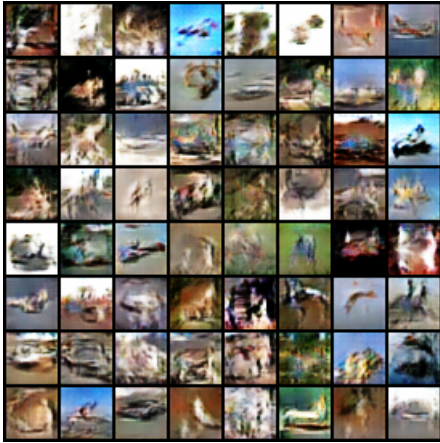


(a) Minibatch Discrimination

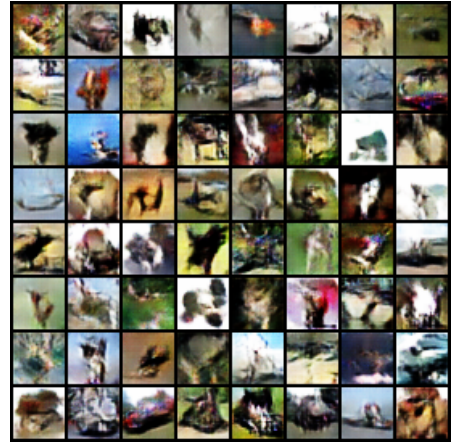


(b) Proposed MC-MBSG

Figure 9: Samples generated by the Generator for the CelebA dataset.



(a) Minibatch Discrimination



(b) Proposed MC-MBSG

Figure 10: Samples generated by the Generator for the CIFAR-10 dataset.

## References

- [1] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 221–231, 2019.
- [2] 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] Chuan Guo, Jacob R. Gardner, Yurong You, A. Wilson, and Kilian Q. Weinberger. Simple black-box adversarial attacks. In *ICML*, 2019.
- [4] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=BkMiWhR5K7>.
- [5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [6] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.
- [7] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.
- [8] Eitan Richardson and Yair Weiss. On gans and gmms. In *Advances in Neural Information Processing Systems*, pages 5847–5858, 2018.
- [9] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL <http://arxiv.org/abs/1606.03498>.
- [10] Chu Wang, Babak Samari, Vladimir G Kim, Siddhartha Chaudhuri, and Kaleem Siddiqi. Affinity graph supervision for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8247–8255, 2020.
- [11] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.
- [12] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.