Supplemental Material for "Subsampling Generative Adversarial Networks: Density Ratio Estimation in Feature Space with Softplus Loss"

Xin Ding, Z. Jane Wang, *Fellow, IEEE* and William J. Welch

S.I. More Details of Simulation in Section IV-A

A. Network Architectures

TABLE S.I.1: Network architectures for the generator and discriminator in the simulation. "fc" denotes a fully-connected layer.

(a) Generator

$z \in \mathbb{R}^{128} \sim N(0, I)$
fc$\rightarrow$ 100; ReLU
fc$\rightarrow$ 100; ReLU
fc$\rightarrow$ 100; ReLU
fc$\rightarrow$ 2

(b) Discriminator

A sample $x \in \mathbb{R}^2$
fc$\rightarrow$ 100; ReLU
fc$\rightarrow$ 100; ReLU
fc$\rightarrow$ 100; ReLU
fc$\rightarrow$ 1; Sigmoid

TABLE S.I.2: Architecture of the 5-layer MLP for DRE in the simulation. We use group normalization (GN) [1] in each hidden layer instead of batch normalization [2] because we find batch normalization performs quite differently in the training stage and evaluation stage.

A sample $x \in \mathbb{R}^2$
fc$\rightarrow$ 2048; GN (4 groups); ReLU; dropout ($p = 0.2$)
fc$\rightarrow$ 1024; GN (4 groups); ReLU; dropout($p = 0.2$)
fc$\rightarrow$ 512; GN (4 groups); ReLU; dropout($p = 0.2$)
fc$\rightarrow$ 256; GN (4 groups); ReLU; dropout($p = 0.2$)
fc$\rightarrow$ 128; GN (4 groups); ReLU; dropout($p = 0.2$)
fc$\rightarrow$ 1; ReLU

B. Training Setups

The GAN model is trained for 50 epochs with the Adam [3] optimizer, a constant learning rate 10^{-3} and batch size 512. DR models are trained with the setups in Table S.I.3, and the hyperparameter selections are in Table S.I.4.

Xin Ding and William J. Welch are with the Department of Statistics, University of British Columbia, Vancouver, BC, V6T 1Z4 Canada (e-mail: xin.ding@stat.ubc.ca, will@stat.ubc.ca) *(Corresponding author: Xin Ding)*.

Jane Z. Wang is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, V6T 1Z4 Canada (e-mail: zjanew@ece.ubc.ca).

Manuscript received March 3, 2020; revised March 3, 2020.

TABLE S.I.3: Setups for training the 5-layer MLP under different loss functions in the simulation.

Loss	uLISF	DSKL	BARR	SP
Optimizer	Adam [4]	Adam [4]	Adam [4]	Adam [4]
Constant LR	1e-5	1e-5	1e-5	1e-3
Epochs	400	400	400	400
Batch Size	512	512	512	512

TABLE S.I.4: Hyperparameter selection in Simulation. Two-Sample Kolmogorov-Smirnov test statistic is shown for each λ at each round.

	Round 1				
λ	0	0.005	0.01	0.05	0.1
KS Stat.	0.00464	0.00388	**0.00386**	0.00496	0.0046
	Round 2				
λ	0	0.005	0.01	0.05	0.1
KS Stat.	0.00896	**0.00838**	0.01044	0.0096	0.01068
	Round 3				
λ	0	0.005	0.01	0.05	0.1
KS Stat.	0.00494	0.00508	**0.00398**	0.008	0.00588

S.II. How does the loss function affect density ratio estimation and the subsequent subsampling?

A. Experimental Setups

We repeat the following steps 5 times and report the average result. For each one, we do the following:

1) Train the GAN for 50 epochs. Train the density ratio model (MLP-5) with the SP loss and uLSIF loss respectively for 5000 epochs. We use the Adam [3] optimizer. The initial learning rate for both loss functions is set to 10^{-3} and decayed every 1000 epochs. Set 17 checkpoints during training from the 20th epoch to the 5000th epoch. The hyperparameter λ is set to 0 or 0.05.
2) Use GMM to model p_g. The optimal number of components is selected by minimizing BIC. Since we can draw a large number of fake samples from p_g, the estimation of p_g by GMM should be very accurate. In this experiment, we use 100,000 fake samples to fit the GMM. The true distribution p_r is a mixture of 25 Gaussian with parameters known to us. Therefore, we can know the ground truth density ratio function $r = p_r/p_g$.
3) At each checkpoint, evaluate the ground truth density ratio function r and two estimated density ratio functions $\hat{r}$ (trained with SP and uLSIF respectively) on all 10000

fake samples, only high-quality fake samples, and only low-quality fake samples. Report the average density ratio of these fake samples at each checkpoint.

4) At each checkpoint, we use DRE-SP+SIR to draw 10,000 fake samples and report the percentage of high quality samples.

B. Results for $\lambda = 0.05$

In Fig. 4 of the main article, we show the results when $\lambda = 0.05$. Note that, in Figure 4a and 4b we only report the training loss of the first round of 5 repetitions; however, in Figure 5a and 5b, the average density ratios of high/low quality samples and the percentage of high quality samples are averaged over 5 repetitions.

In Fig. S.II.1, we also show the average density ratio over all 10,000 fake samples and the percentage of high quality samples (averaged over 5 repetitions). We can see, when using the SP loss, the average density ratio over all 10,000 fake samples is close to the ground truth. However, when using the uLSIF loss, the average density ratio keeps decreasing (a sign of overfitting).

In Fig. 4a, the training loss of uLSIF stops decreasing after 3000 epochs because of a too small learning rate which seems to conflict with our argument in Section II-D. In Fig. S.II.2, however, we show the training curve for a constant learning rate 10^{-5} and we can see a constantly decreasing trend.

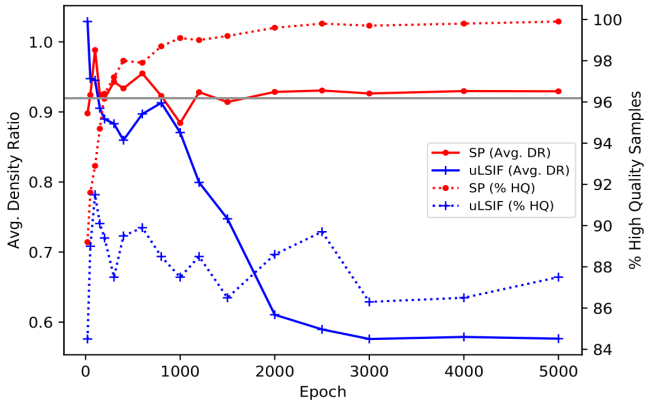


Fig. S.II.1: Average density ratio of 10,000 fake samples and percentage of high quality samples versus the epoch of DRE training when $\lambda = 0.05$. The grey horizontal line stands for the ground truth average density ratio.

C. Results for $\lambda = 0$

Besides $\lambda = 0.05$, we also set λ to 0 and show the corresponding results in Figure S.II.3 and S.II.4. From Fig. S.II.3, we can see both loss functions suffer from overfitting. However, the overfitting problem for the uLSIF loss is more obvious than the SP loss, because Fig. S.II.4 shows that the difference between high quality and low quality samples in terms of the average density ratios is bigger when using SP loss. Therefore, even with the overfitting problem, the SP loss still leads to a much better subsampling performance than the uLSIF loss does.

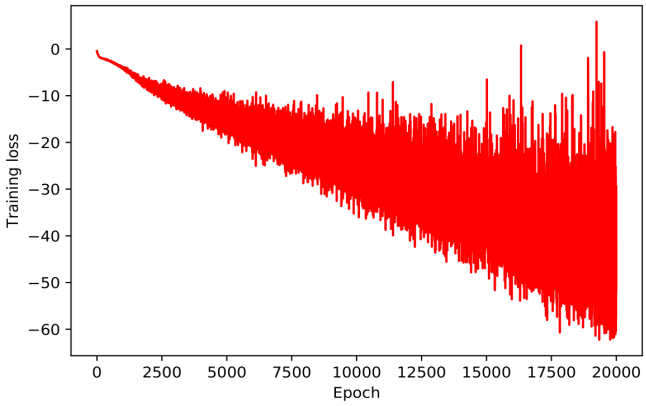


Fig. S.II.2: The training curve of a 5-layer MLP under the penalized uLSIF loss (12) with $\lambda = 0.05$ and a constant learning rate 10^{-5}.

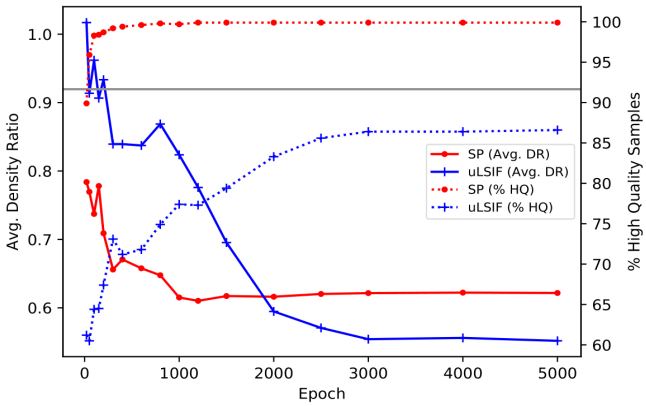


Fig. S.II.3: Average density ratio of 10,000 fake samples and percentage of high quality samples versus the epoch of DRE training when $\lambda = 0$. The grey horizontal line stands for the ground truth average density ratio.

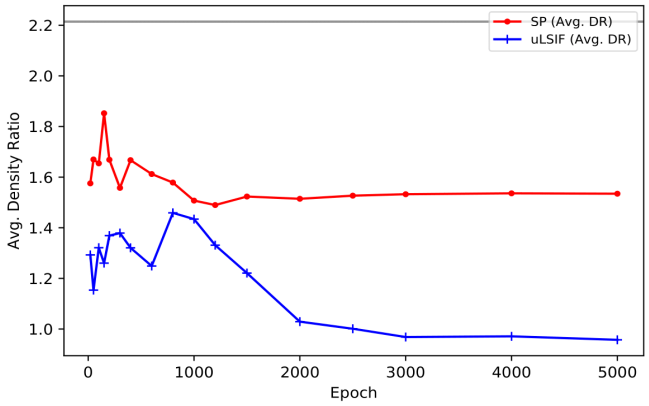
S.III. Real Datasets

A. Evaluation Metrics

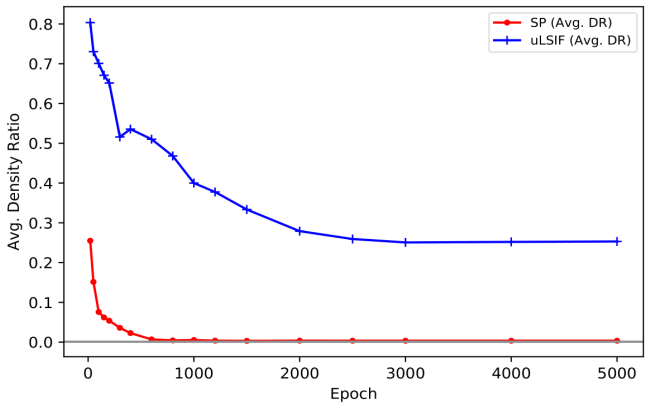
Inception Score (IS) [5] is a popular evaluation metric for GANs which is defined as follows:

$$IS = \exp\{\mathbb{E}_{\boldsymbol{x}}[\mathbb{KL}(p(y|\boldsymbol{x})\|p(y))]\} = \exp\{H(y) - \mathbb{E}_{\boldsymbol{x}}[H(y|\boldsymbol{x})]\}, \tag{S.1}$$

where $p(y|\boldsymbol{x})$ is the conditional label distribution for an image $\boldsymbol{x}$, $p(y)$ is the marginal label distribution, $H(y)$ is the entropy of y, and $H(y|\boldsymbol{x})$ is the entropy of y conditioning on $\boldsymbol{x}$. The distribution $p(y|\boldsymbol{x})$ is often modeled by a pre-trained CNN, say Inception-V3 [6], and $p(y) \approx (1/N)\sum_{i=1}^{N} p(y|\boldsymbol{x}_i)$. IS evaluates the quality of a set of fake images from two perspectives: high classifiability and diversity with respect to class labels. We assume high quality images are more classifiable so we favor smaller $\mathbb{E}_{\boldsymbol{x}}[H(y|\boldsymbol{x})]$. On the other hand, high diversity means the GAN model can generate images from all potential classes instead of a few classes so we expect high entropy in the class labels (predicted by the pre-trained Inception-V3) of those fake images (i.e., larger $H(y)$). Therefore, the larger the IS is, the better quality the fake images have.



(a) Average density ratio of high quality fake samples versus the epoch of DRE training when $\lambda = 0$.



(b) Average density ratio of low quality fake samples versus the epoch of DRE training when $\lambda = 0$.

Fig. S.II.4: Average density ratio of high/low quality fake samples versus the epoch of DRE training when $\lambda = 0$. The grey horizontal line stands for the ground truth.

Fréchet Inception Distance (FID) [7] is another popular evaluation metric for GAN models. The FID is defined on a feature space learned by a pre-trained CNN and we assume a feature $\boldsymbol{y}$ extracted by this pre-trained CNN follows a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. In other words, we assume $\boldsymbol{y}^r \sim \mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ and $\boldsymbol{y}^g \sim \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$, where $\boldsymbol{y}^r$ and $\boldsymbol{y}^g$ are extracted features of real and fake images respectively. This assumption looks very strong, but empirical studies show that FID is consistent with human judgments and works more robustly than IS does [7]. FID is defined as follows

$$FID = \|\boldsymbol{\mu}_r - \boldsymbol{\mu}_g\|_2^2 + \mathrm{Tr}(\boldsymbol{\Sigma}_r + \boldsymbol{\Sigma}_g - 2(\boldsymbol{\Sigma}_r\boldsymbol{\Sigma}_g)^{\frac{1}{2}}), \quad \text{(S.2)}$$

where $\boldsymbol{\mu}_r$, $\boldsymbol{\mu}_g$, $\boldsymbol{\Sigma}_g$ and $\boldsymbol{\Sigma}_r$ can be estimated from samples. Note that FID is computed based on both real images and fake images while IS is only computed on fake images.

B. CIFAR-10

1) Network Architectures: We implement DCGAN and WGAN-GP with generator and discriminator shown in Table S.III.1. For MMD-GAN, we directly use codes in https://github.com/OctoberChang/MMD-GAN. Please see [8] for more details about MMD-GAN.

TABLE S.III.1: Network architectures for the generator and discriminator of DCGAN and WGAN-GP in the experiment on CIFAR-10. The slopes of all LeakyReLU are set to 0.2. We denote stride and padding by s and p respectively.

(a) Generator

$z \in \mathbb{R}^{128} \sim N(0, I)$
fc$\rightarrow 4 \times 4 \times 512$
deconv, 4×4, s $= 2$, p $= 1$, 256; BN; ReLU
deconv, 4×4, s $= 2$, p $= 1$, 128; BN; ReLU
deconv, 4×4, s $= 2$, p $= 1$, 64; BN; ReLU
conv, 3×3, s $= 1$, p $= 1$, 3; Tanh

(b) Discriminator

RGB image $x \in \mathbb{R}^{3\times32\times32}$
conv, 3×3, s $= 1$, p $= 1$, 64; LeakyReLU
conv, 4×4, s $= 2$, p $= 1$, 64; LeakyReLU
conv, 3×3, s $= 1$, p $= 1$, 128; LeakyReLU
conv, 4×4, s $= 2$, p $= 1$, 128; LeakyReLU
conv, 3×3, s $= 1$, p $= 1$, 256; LeakyReLU
conv, 4×4, s $= 2$, p $= 1$, 256; LeakyReLU
conv, 3×3, s $= 1$, p $= 1$, 512; LeakyReLU
fc$\rightarrow 1$
sigmoid (for DCGAN only)

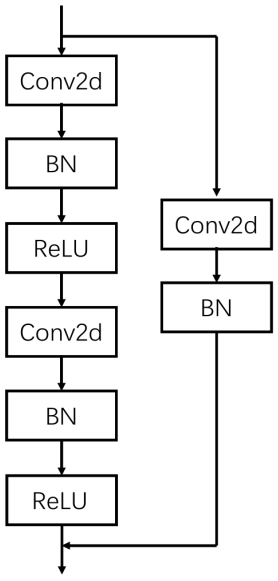


Fig. S.III.1: A residual block (ResBlock) in ResNet-34 for CIFAR-10. Please check our codes for detailed setups of each layer.

TABLE S.III.2: Architecture of the ResNet-34 for feature extraction for CIFAR-10. "down" refers to down sampling. We add an extra fully-connected layer fc1 to output features for density ratio estimation in feature space. $\times n$ represents n consecutive such blocks.

RGB image $x \in \mathbb{R}^{3\times32\times32}$
conv, 3x3, stride=1, p=1, 64; BN; ReLU
{ResBlock, 64} $\times3$
ResBlock, down, 128
{ResBlock, 128}$\times3$
ResBlock, down, 256
{ResBlock, 256} $\times5$
ResBlock, down, 512
{ResBlock, 512} $\times2$
Avg. pooling, 4×4, s $= 4$
fc1$\rightarrow 32 \times 32 \times 3 = 3072$
fc2$\rightarrow 10$

TABLE S.III.3: 5-layer MLP for DRE in feature space for CIFAR-10.

extracted feature $\boldsymbol{y} \in \mathbb{R}^{3072}$
fc$\rightarrow$ 2048, GN (4 groups), ReLU, Dropout($p = 0.4$)
fc$\rightarrow$ 1024, GN (4 groups), ReLU, Dropout($p = 0.4$)
fc$\rightarrow$ 512, GN (4 groups), ReLU, Dropout($p = 0.4$)
fc$\rightarrow$ 256, GN (4 groups), ReLU, Dropout($p = 0.4$)
fc$\rightarrow$ 128, GN (4 groups), ReLU, Dropout($p = 0.4$)
fc$\rightarrow$ 1, ReLU

TABLE S.III.4: 2-layer CNN for DRE in pixel space [9] for CIFAR-10

RGB image $\boldsymbol{x} \in \mathbb{R}^{3\times32\times32}$
conv, 9×9, $s = 1$, 6
Avg. pooling, 2×2, $s = 2$
Sigmoid
conv, 9×9, $s = 1$, 12
Avg. pooling, 2×2, $s = 2$
Sigmoid
fc$\rightarrow$ 1; ReLU

TABLE S.III.5: 6-layer CNN for DRE in pixel space [10] for CIFAR-10

RGB image $\boldsymbol{x} \in \mathbb{R}^{3\times32\times32}$
conv, 3×3, $s = 1$, 60
Max pooling, 2×2, $s = 1$
conv, 3×3, $s = 1$, 50
conv, 3×3, $s = 1$, 40
Max pooling, 2×2, $s = 1$
conv, 3×3, $s = 1$, 20
Max pooling, 2×2, $s = 1$
conv, 2×2, $s = 1$, 10
conv, 2×2, $s = 1$, 5
fc$\rightarrow$ 250, ReLU, Dropout($p = 0.25$)
fc$\rightarrow$ 1; ReLU

TABLE S.III.6: Binary classifier for the BOC-based DRE [11] for CIFAR-10

RGB image $\boldsymbol{x} \in \mathbb{R}^{3\times32\times32}$
conv, 3×3, $s = 1$, $p = 1$, 64; ReLU; BN
Max pooling, 2×2, $s = 2$
conv, 3×3, $s = 1$, $p = 1$, 64; ReLU; BN
Max pooling, 2×2, $s = 2$
conv, 3×3, $s = 1$, $p = 1$, 64; ReLU; BN
Max pooling, 2×2, $s = 2$
conv, 3×3, $s = 1$, $p = 1$, 64; ReLU; BN
Max pooling, 2×2, $s = 2$
fc$\rightarrow$ 1; Sigmoid

2) Training Setups: In the CIFAR-10 setting, three types of GAN are trained on the 50,000 training images with setups in Table S.III.7. The modified ResNet-34 for feature extraction is trained on the training set for 200 epochs with the SGD optimizer, initial learning rate 0.1 (decay at epoch 100 and 150 with factor 0.1), weight decay 10^{-4}, and batch size 256. The training setups for different DRE methods are shown in Table S.III.8.

TABLE S.III.7: Training setups for three types of GAN on CIFAR-10.

	DCGAN	WGAN-GP	MMD-GAN
Optimizer	Adam	Adam	Adam
Constant learning rate	2E-04	2E-04	5E-05
Epochs	500	2000	4000
Batch Size	256	256	256

TABLE S.III.8: Training setups for DRE methods on CIFAR-10. Corresponding subsampling results are shown in Table III to V of the main article.

	DRE-P-uLSIF	DRE-P-DSKL	DRE-P-BARR	BOC
Optimizer	Adam	Adam	Adam	Adam
Initia LR	1E-4	1E-4	1E-4	1E-3
LR Decay	No	No	No	No
Epochs	200	200	200	100
Batch Size	512	512	512	100
	DRE-F-uLSIF	DRE-F-DSKL	DRE-F-BARR	DRE-F-SP
Optimizer	Adam	Adam	Adam	Adam
Initia LR	1E-5	1E-5	1E-5	1E-4
LR Decay	No	No	No	epoch 100 ($\times0.1$)
Epochs	200	200	200	200
Batch Size	512	512	512	512

TABLE S.III.9: Hyperparameter selection for CIFAR-10. Two-Sample Kolmogorov-Smirnov test statistic is shown for each λ and each GAN.

			DCGAN		
λ	0	0.005	0.01	0.05	0.1
KS Stat.	**1.380E-01**	1.390E-01	1.386E-01	1.389E-01	1.394E-01
			WGAN-GP		
λ	0	0.005	0.01	0.05	0.1
KS Stat.	1.131E-01	**1.118E-01**	1.138E-01	1.165E-01	1.204E-01
			MMD-GAN		
λ	0	0.005	0.006	0.008	0.01
KS Stat.	1.220E-01	1.231E-01	**1.209E-01**	1.233E-01	1.209E-01

3) Performance Measures: To evaluate the quality of fake images by IS and FID, we train the Inception-V3 on 50,000 CIFAR-10 training images. FID is computed based on the final average pooling features from the pre-trained Inception-V3. We compute the FID between 50,000 fake images and 50,000 training images.

4) Visual Results: We show some example CIFAR-10 images from different subsampling methods in Fig. S.III.2 to S.III.4. For each method, we draw images from two classes (car and horse) with 50 images per class (first 5 rows correspond to cars and the rest correspond to horses). The improvement of our methods is more obvious on images from WGAN-GP in Fig. S.III.3 where our methods generate more recognizable cars and horses.

C. CIFAR-10: An Extra Experiment

[13] compares the authors' MH-GAN with DRS [12] and no subsampling when the DCGAN is trained for only 60 epochs. To show how the epoch of DCGAN training influences the subsampling results, we fix all settings (e.g., $\lambda = 0$) but only vary the epoch of DCGAN training and visualize the results in Fig. S.III.5. In terms of IS and FID, our proposed DRE-F-SP+RS is consistently better than other methods at all epochs.

D. Reduced MNIST

1) Data: The MNIST dataset has 70,000 28×28 gray-scale handwritten digits from 10 classes (10 digits). The dataset is split into a training set of 60,000 images with 6000 per class and a test set of 10,000 images with 1000 per class. Since MNIST is a very simple dataset, to increase the difficulty for both the GAN training and the subsampling, we randomly select 5000 images from the original training set to form a reduced training set.

2) Network Architectures: We implement DCGAN with the generator and the discriminator shown in Table S.III.10. The ResNet-34 for feature extraction and the MLP-5 for DRE are identical to the CIFAR-10 experiment except a few setups (e.g., the number of input channels of the first convolutional layer of ResNet-34) and please see our codes for more details.

TABLE S.III.10: Network architectures for the generator and the discriminator of DCGAN in the experiment on the MNIST. The slopes of all LeakyReLU are set to 0.2. We denote stride and padding by s and p respectively.

(a) Generator

$z \in \mathbb{R}^{128} \sim N(0, I)$
dense$\rightarrow 7 \times 7 \times 256$
deconv, 4×4, s $= 2$, p $= 1$, 128; BN; ReLU
deconv, 4×4, s $= 2$, p $= 1$, 64; BN; ReLU
conv, 3×3, s $= 1$, p $= 1$, 3; Tanh

(b) Discriminator

RGB image $x \in \mathbb{R}^{1 \times 28 \times 28}$
conv, 3×3, s $= 1$, p $= 1$, 64; LeakyReLU
conv, 4×4, s $= 2$, p $= 1$, 64; LeakyReLU
conv, 3×3, s $= 1$, p $= 1$, 128; LeakyReLU
conv, 4×4, s $= 2$, p $= 1$, 128; LeakyReLU
conv, 3×3, s $= 1$, p $= 1$, 256; LeakyReLU
fc$\rightarrow 1$; Sigmoid

3) Training Setups: In the MNIST setting, the DCGAN is trained on the reduced training set with the Adam optimizer, constant learning rate 10^{-4}, batch size 256, and 500 epochs. The modified ResNet-34 for feature extraction is trained on the reduced training set for 200 epochs with the SGD optimizer, initial learning rate 0.01 (decay at epoch 100 with factor 0.1), weight decay 10^{-4}, and batch size 512. To implement DRE-F-SP, we use the MLP-5 as in the CIFAR-10 setting, and it is trained with the Adam optimizer, initial learning rate 10^{-4} (decay at the epoch 400 with factor 0.1), batch size 512, and 500 epochs. The optimal hyperparameter is selected based on Table S.III.11

TABLE S.III.11: Hyperparameter selection for MNIST. Two-Sample Kolmogorov-Smirnov test statistic is shown for each λ.

	DCGAN			
λ	0	0.001	0.01	0.1
KS Stat.	1.203E-01	1.182E-01	1.168E-01	**1.115E-01**

4) Performance Measures: To evaluate the quality of fake images by IS and FID, we train the Inception-V3 on all 60,000 MNIST training images (not the reduced training set). FID is computed based on the final average pooling features from the pre-trained Inception-V3. We compute the FID between 50,000 fake images and all 60,000 training images.

5) Quantitative Results: Table S.III.12 shows the results of the experiment and demonstrates that our approaches significantly outperform two existing subsampling methods.

TABLE S.III.12: Average quality of 50,000 fake MNIST images from different subsampling methods over three repetitions. We draw 50,000 fake images by each method on which we compute the IS and FID. We repeat this sampling for three times and report the average IS and FID. Higher IS and lower FID are better. A grid search is conducted for DRE-F-SP to select the hyperparameter, and the results under the optimal λ^* are shown in this table. We include the IS and FID of all 60,000 training data and 10,000 test data as a reference.

Method	IS (mean$\pm$std)	FID (mean$\pm$std)
- Real Data -		
All 60,000 Training Data	9.984	-
10,000 Test Data	9.462	0.134
- DCGAN -		
No Subsampling	7.791 ± 0.004	1.723 ± 0.004
DRS [12]	8.032 ± 0.007	1.658 ± 0.007
MH-GAN [13]	8.461 ± 0.001	1.269 ± 0.007
DRE-F-SP+RS ($\lambda^* = 0.1$)	$\mathbf{9.676 \pm 0.005}$	$\mathbf{0.374 \pm 0.007}$
DRE-F-SP+MH ($\lambda^* = 0.1$)	$\mathbf{9.677 \pm 0.006}$	$\mathbf{0.368 \pm 0.008}$
DRE-F-SP+SIR ($\lambda^* = 0.1$)	$\mathbf{9.675 \pm 0.003}$	$\mathbf{0.380 \pm 0.010}$

6) Visual Results: We show some example MNIST images from different subsampling methods in Fig. S.III.6. For each method, we draw images from all 10 classes with 30 images per class.

E. CelebA Dataset

1) Data: The CelebA dataset [14] has 202,599 218×178 RGB face images. Each image has 40 binary attributes. We create 6 classes in terms of 3 binary attributes ("Wearing_Lipstick", "Smiling", "Mouth_Sligthly_Open"); some minority classes are merged together to balance the dataset. Then

the dataset is randomly split into a training set of 192,599 images and a test set of 10,000 images. All images are resized to 64×64.

2) Network Architectures: We implement SNGAN [15] based on codes from https://github.com/christiancosgrove/pytorch-spectral-normalization-gan and https://github.com/pfnet-research/sngan_projection. The ResNet-34 for feature extraction and the MLP-5 for DRE are identical to the CIFAR-10 experiment except a few setups (e.g., the stride size of the first residual block of ResNet-34) and please see our codes for more details.

3) Training Setups: The SNGAN is trained on the training set with the Adam optimizer, constant learning rate 10^{-4} for the generator, constant learning rate 4×10^{-4} for the discriminator, batch size 256, and 100 epochs. The modified ResNet-34 for feature extraction is trained on the training set for 100 epochs with the SGD optimizer, initial learning rate 0.001 (decay at epoch 30 and 70 with factor 0.1), weight decay 10^{-4}, and batch size 256. The MLP-5 is trained on the training set with the Adam optimizer, initial learning rate 10^{-4} (decay at epoch 30 and 70), batch size 512, and 100 epochs. The optimal hyperparameter is selected based on Table S.III.13.

TABLE S.III.13: Hyperparameter selection for CelebA. Two-Sample Kolmogorov-Smirnov test statistic is shown for each λ.

	SNGAN				
λ	0	**0.005**	0.01	0.05	0.1
KS Stat.	0.2643	**0.2622**	0.2646	0.2643	0.2645

4) Performance Measures: To evaluate the quality of fake images by IS and FID, we use the Inception-V3 which is pre-trained on the ImageNet [16]. FID is computed based on the final average pooling features from the pre-trained Inception-V3. The computation is based on codes from https://github.com/sbarratt/inception-score-pytorch and https://github.com/mseitzer/pytorch-fid. We compute the FID between 50,000 fake images and all 192,599 training images.

5) Quantitative Results: Table S.III.14 shows the results of the experiment and demonstrates that our approaches significantly outperform two existing subsampling methods.

6) Visual Results: We show some example CelebA images from different subsampling methods in Fig. S.III.7. We manually check each generated image and outline it in yellow if we find obvious distortion or asymmetry in its face area (we ignore the background). We can see our proposed methods have fewer marked images.

TABLE S.III.14: Average quality of 50,000 fake CelebA images from different subsampling methods over three repetitions. We draw 50,000 fake images by each method on which we compute the IS and FID. We repeat this sampling for three times and report the average IS and FID. Higher IS and lower FID are better. A grid search is conducted for DRE-F-SP to select the hyperparameter, and the results under the optimal λ^* are shown in this table. We include the IS and FID of all training data and 10,000 test data as a reference.

Method	IS (mean±std)	FID (mean±std)
- Real Data -		
All Training Data	3.321	-
10,000 Test Data	3.319	1.572
- SNGAN -		
No Subsampling	2.781 ± 0.006	6.607 ± 0.013
DRS [12]	2.796 ± 0.006	6.490 ± 0.015
MH-GAN [13]	2.766 ± 0.005	6.649 ± 0.019
DRE-F-SP+RS ($\lambda^* = 0.005$)	$\mathbf{2.865 \pm 0.009}$	$\mathbf{6.088 \pm 0.007}$
DRE-F-SP+MH ($\lambda^* = 0.005$)	$\mathbf{2.870 \pm 0.009}$	$\mathbf{6.032 \pm 0.019}$
DRE-F-SP+SIR ($\lambda^* = 0.005$)	$\mathbf{2.877 \pm 0.021}$	$\mathbf{6.116 \pm 0.021}$

REFERENCES

[1] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[2] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[3] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15, 2011, pp. 215–223.

[4] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[5] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 2234–2242.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[7] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.

[8] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Poczos, "MMD GAN: Towards deeper understanding of moment matching network," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 2203–2213.

[9] H. Nam and M. Sugiyama, "Direct density ratio estimation with convolutional neural networks with application in outlier detection," *IEICE Transactions on Information and Systems*, vol. 98, no. 5, pp. 1073–1079, 2015.

[10] H. Khan, L. Marcuse, and B. Yener, "Deep density ratio estimation for change point detection," *arXiv preprint arXiv:1905.09876*, 2019.

[11] A. Grover, J. Song, A. Agarwal, K. Tran, A. Kapoor, E. Horvitz, and S. Ermon, "Bias correction of learned generative models using likelihood-free importance weighting," *arXiv preprint arXiv:1906.09531*, 2019.

[12] S. Azadi, C. Olsson, T. Darrell, I. Goodfellow, and A. Odena, "Discriminator rejection sampling," in *International Conference on Learning Representations*, 2019.

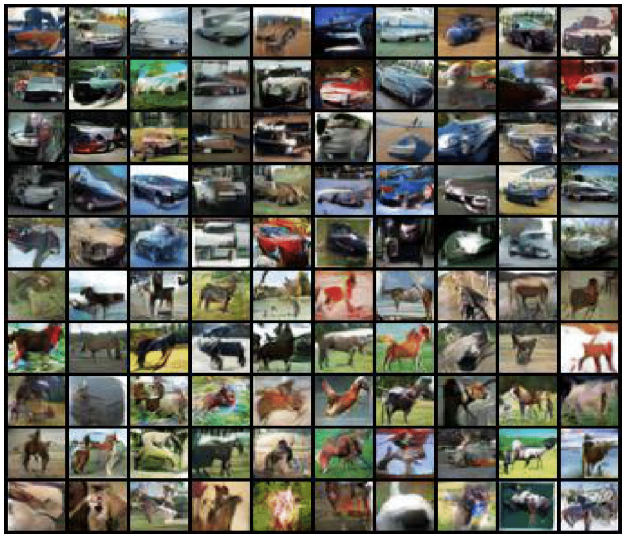
[13] R. Turner, J. Hung, E. Frank, Y. Saatchi, and J. Yosinski, "Metropolis-Hastings generative adversarial networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 6345–6353.

[14] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

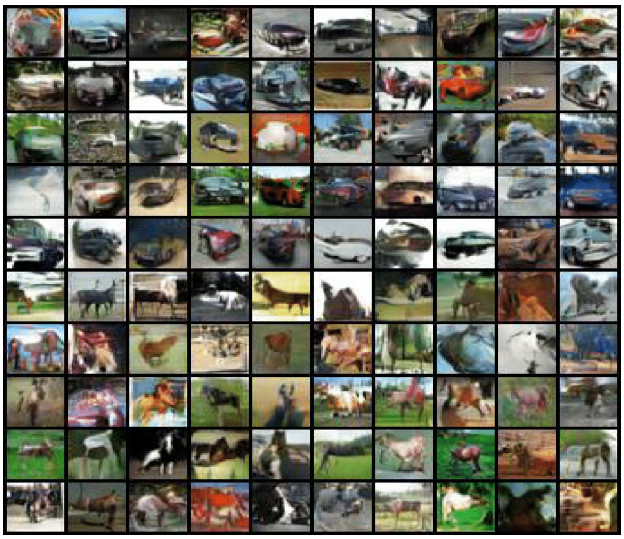
[15] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *CVPR09*, 2009.

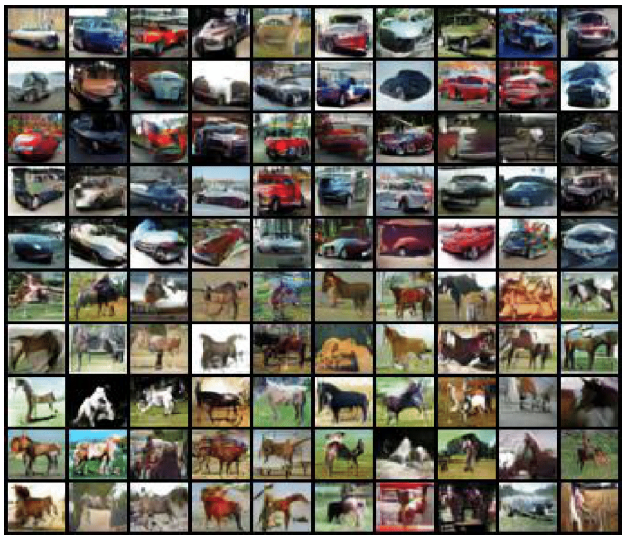
(a) No subsampling



(b) DRS [12]

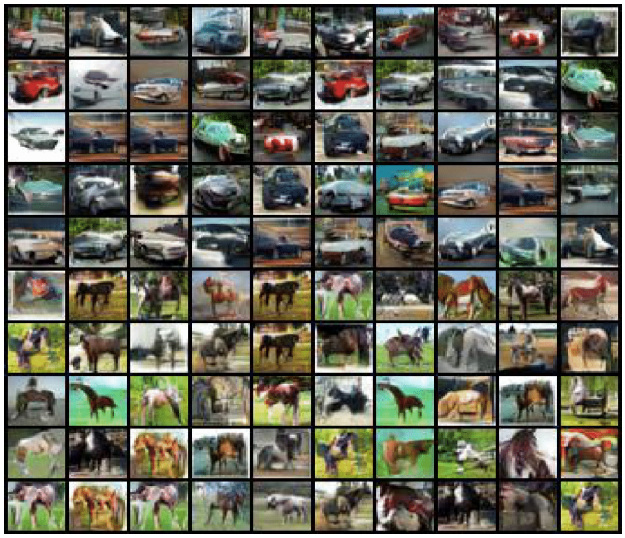


(c) MH-GAN [13]



(d) DRE-F-SP+RS

(e) DRE-F-SP+MH



(f) DRE-F-SP+SIR

Fig. S.III.2: Fake CIFAR-10 images (car and horse) from DCGAN

.

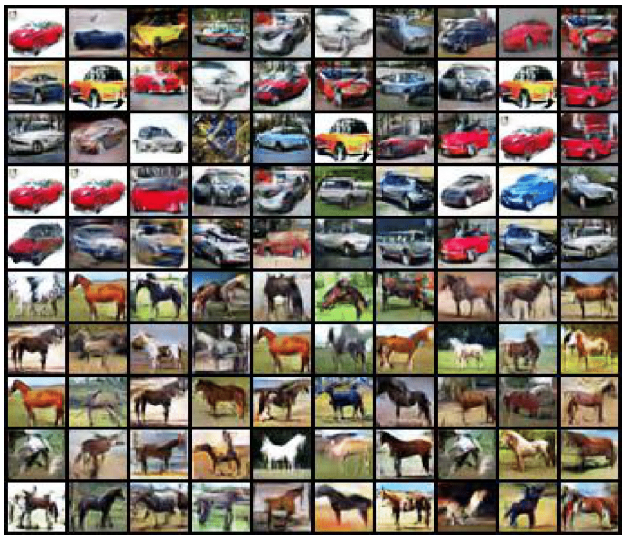
(a) No subsampling

(b) DRS [12]

(c) MH-GAN [13]

(d) DRE-F-SP+RS

(e) DRE-F-SP+MH



(f) DRE-F-SP+SIR

Fig. S.III.3: Fake CIFAR-10 images (car and horse) from WGAN-GP.

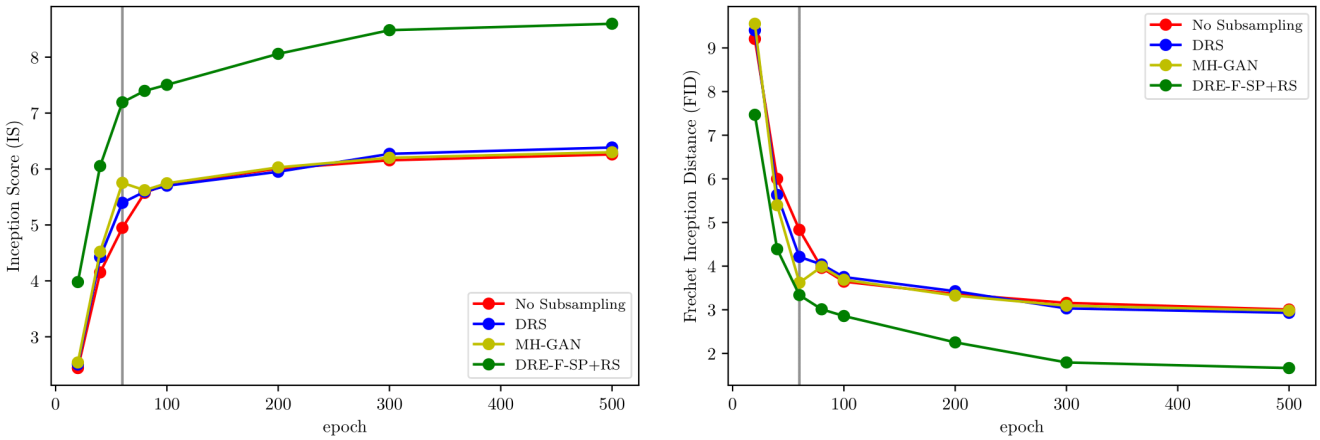
(a) No subsampling

(b) DRE-F-SP+RS

(c) DRE-F-SP+MH

(d) DRE-F-SP+SIR

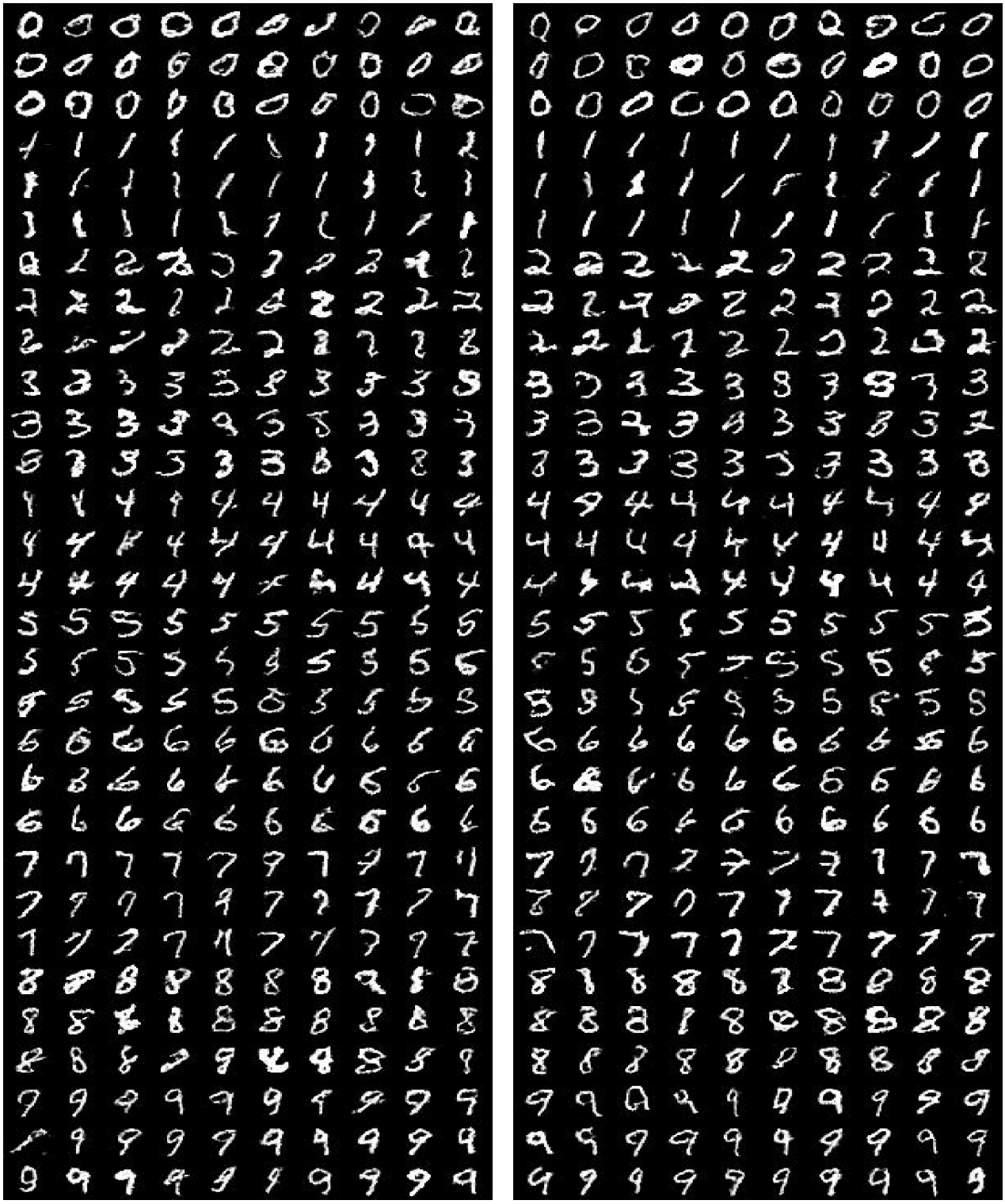
Fig. S.III.4: Fake CIFAR-10 images (car and horse) from MMD-GAN.



(a) IS versus the epoch of GAN training

(b) FID versus the epoch of GAN training

Fig. S.III.5: Inception Score (IS) and Frechet Inception Distance (FID) versus the epoch of DCGAN training. The grey vertical line indicates the epoch 60. DRS [12] and MH-GAN [13] perform well relative to the baseline of No Subsampling only at epoch 60 but the DCGAN is not well-trained at that epoch. It is more reasonable to train the DCGAN well first (e.g., 500 epochs) and then apply any subsampling method to further improve the image quality.



(a) No subsampling (b) DRS [12]

Fig. S.III.6: Fake MNIST images from DCGAN: Part 1

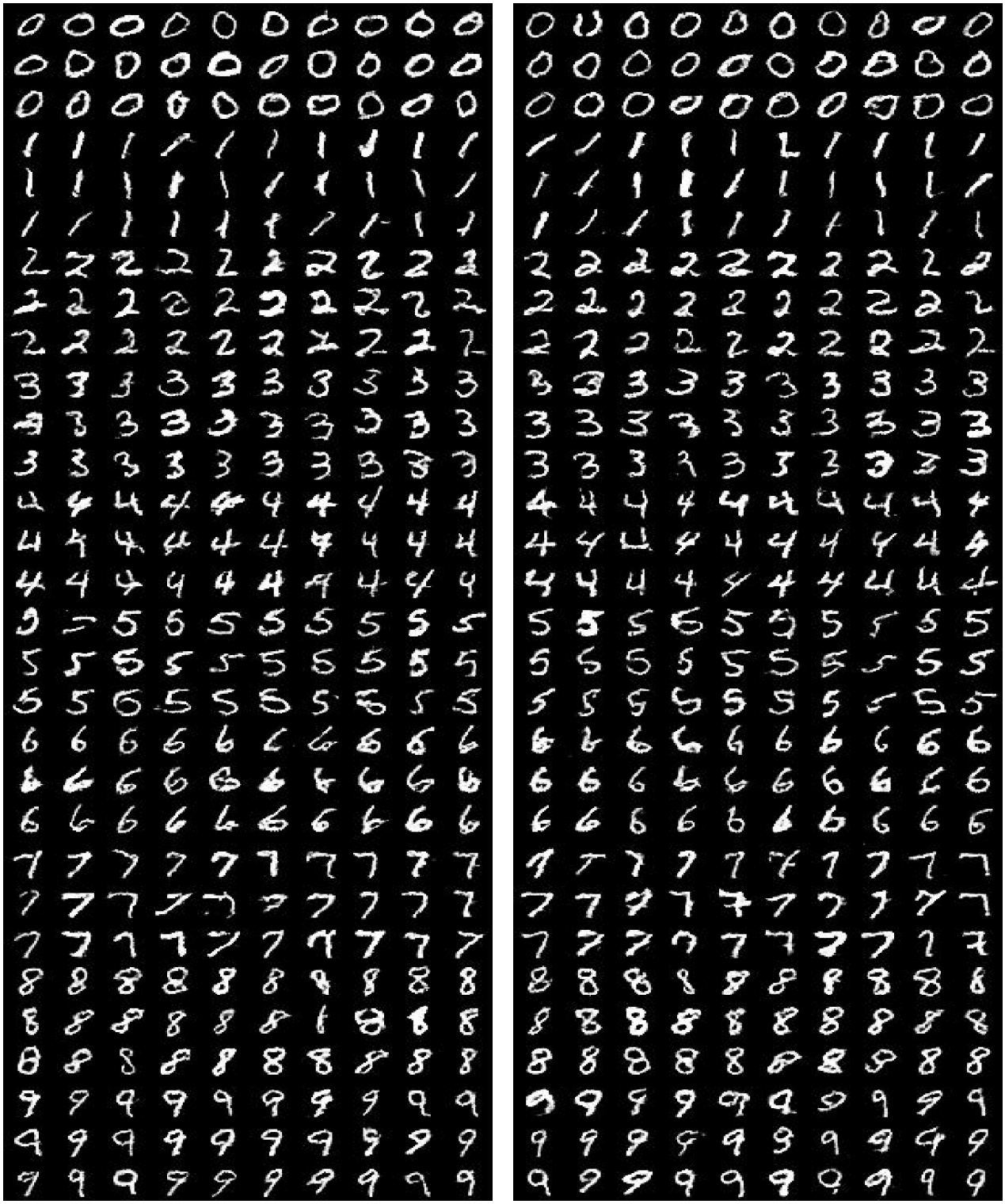
.

(c) MH-GAN [13]

(d) DRE-F-SP+RS

Fig. S.III.6: Fake MNIST images from DCGAN: Part 2

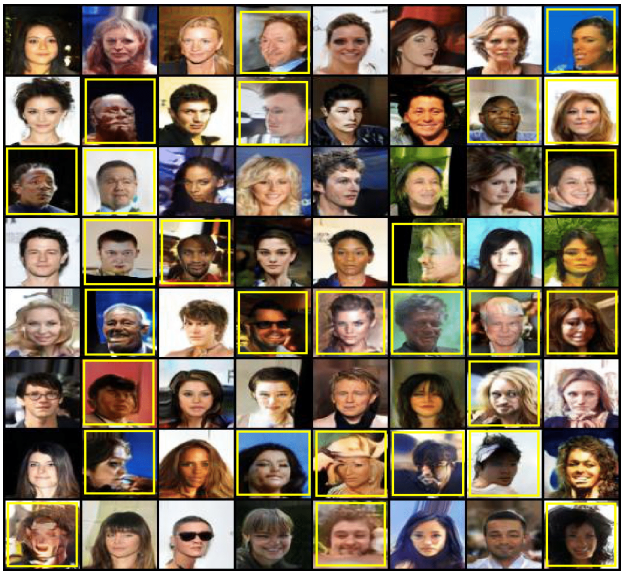
.



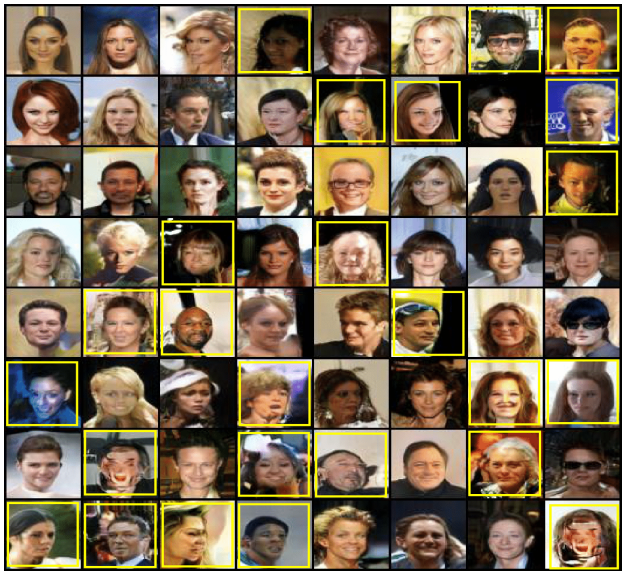
(e) DRE-F-SP+MH (f) DRE-F-SP+SIR

Fig. S.III.6: Fake MNIST images from DCGAN: Part 3

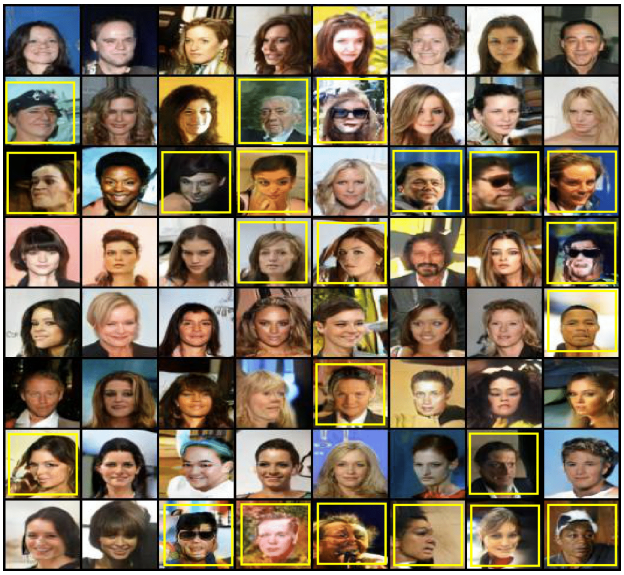
.



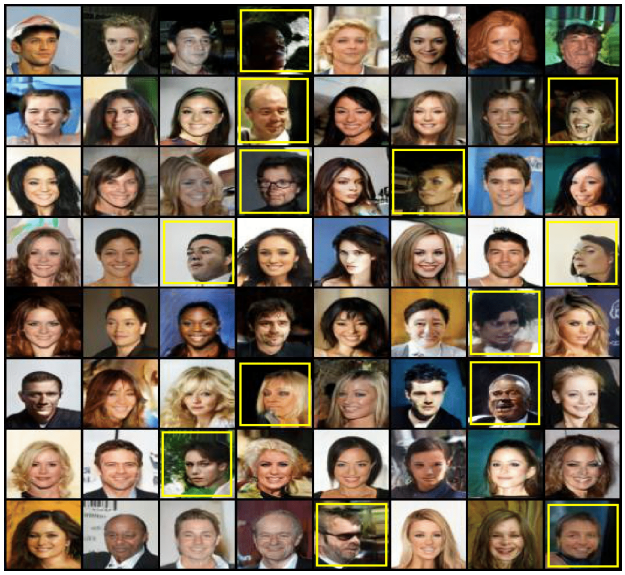
(a) No subsampling (28 yellow squares)



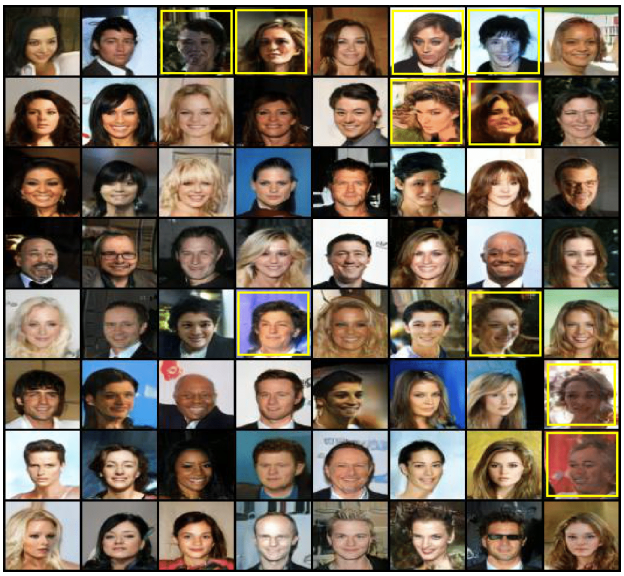
(b) DRS [12] (25 yellow squares)



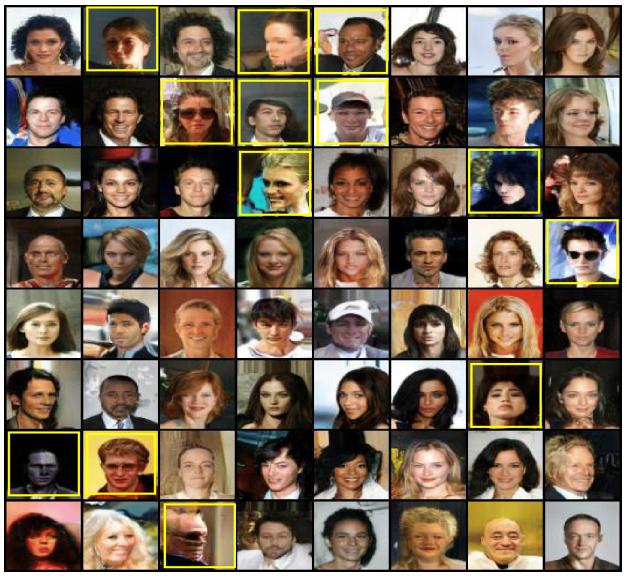
(c) MH-GAN [13] (22 yellow squares)



(d) DRE-F-SP+RS (13 yellow squares)



(e) DRE-F-SP+MH (10 yellow squares)



(f) DRE-F-SP+SIR (13 yellow squares)

Fig. S.III.7: Fake CelebA images from SNGAN (yellow outlines indicate samples with obvious distortions and asymmetry)

.