Inclusive GAN: Improving Data and Minority Coverage in Generative Models (Supplementary Material)

Ning Yu[1,2], Ke Li[3,5,6], Peng Zhou[1]
Jitendra Malik[3], Larry Davis[1], and Mario Fritz[4]

[1] University of Maryland, College Park, United States
[2] Max Planck Institute for Informatics, Saarbrücken, Germany
[3] University of California, Berkeley, United States
[4] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
[5] Institute for Advanced Study, Princeton, United States
[6] Google, Seattle, United States
`ningyu@mpi-inf.mpg.de` `ke.li@eecs.berkeley.edu` `pengzhou@cs.umd.edu`
`malik@eecs.berkeley.edu` `lsd@cs.umd.edu` `fritz@cispa.saarland`

1 Implementation Details

IMLE-GAN. We train each model using Adam optimizer [4] for $K = 300$ epochs. We use no exponential decay rate ($\beta_1 = 0.0$) for the first moment estimates, and use the exponential decay rate $\beta_2 = 0.99$ for the second moment estimates. The learning rate $\eta = 0.002$, the same as in StyleGAN2 [3]. We update the matching of latent vectors to data points every $S = 20$ epochs. The size of the pool of latent vector candidates is 10 times of the size of the dataset or the minority group depending on the application. Perturbation variance is $\sigma^2 = 0.05^2$. The weight of the reconstruction loss varies according to the choice of metric, such that the magnitude of the reconstruction loss is about equal to that of the adversarial loss. For ℓ_2 distance in pixel space, $\lambda = 36$. In the discriminator feature space [5] $\lambda = 9.6 \times 10^6$. In the Inception feature space [2] $\lambda = 10$. For LPIPS [16] $\lambda = 2.5$. Consistently, the weight of the interpolation loss is always set to $\beta = 0.4\lambda$.

We train all our models on 3 NVIDIA V100 Tensor Core GPUs with 16GB memory each. Based on the memory available and the training performance, we set the batch size at 32 for the 240,000 32×32×3 Stacked MNIST images [8], and the training lasts for 1.7 days. We set the batch size at 16 for the 30,000 128×128×3 CelebA images [7], and the training lasts for 2.4 days.

Baseline methods. For fair comparisons, all the baseline methods are reimplemented using the same StyleGAN2 backbone and training strategies. For ALI [1], VAEGAN [5], α-GAN [10], and VEEGAN [13] where an encoder is involved, we adapt the discriminator architecture for the encoder. For DistGAN [14], we measure image distance by LPIPS [16] and tune the weight of the distance constraint term such that its value is about 1/4 of the adversarial loss. For DSGAN [15], we tune the weight of the diversity regularization term such that its value is about 1/4 of the adversarial loss. For PacGAN [6], we set the pack size to 8. For VAEGAN [5], we tune the weights of the data reconstruction term and the prior term such that the former is about equal to the adversarial loss and the latter is about 1/4 of that. For α-GAN [10], we use LPIPS distance [16] to reconstruct images and tune the weight of the reconstruction term such that its value is about 1/4 of the adversarial loss. For VEEGAN [13], we tune the weight of the latent reconstruction term such that its value is about equal to the adversarial loss. For SNGAN [9] and ALI [1], there is no additional hyperparameter.

Evaluation. For Precision and Recall [11] measurement, we use the default setting from their official code repository. In particular, the features are extracted from the *Pool3* layer of a pre-trained Inception network [2]. The number of clusters for k-means is set to 20. We launched for 10 independent runs and report the average for Precision and Recall. For IvOM [8] measurement, the retrieval is implemented as an optimization w.r.t. the latent vector, such that a learned generator approximates its generation towards the query image. The retrieval error is then calculated as the difference between the optimal generated image and the query image. The optimization objective and the error are measured using the deep similarity metric LPIPS [16]. Given each query image and a learned generative model, we optimized the latent vector via Adam [4] for 400 steps. The learning rate setting strategy is the same as in StyleGAN2: the maximum learning rate is 0.1, and it is ramped up from zero linearly during the first 20 steps and ramped down to zero using a cosine schedule during the last 100 steps.

2 Effectiveness of Harmonization

In Section 3.3 in the main paper, we propose two strategies to harmonize adversarial and reconstructive training: the deep distance metric and the interpolation-based augmentation. We compare four distance metrics and with/without augmentation in the third part of Table 1. For distance metrics, the pixel space (the vanilla version) achieves the desirable Recall and the Inception space achieves the desirable FID, but they contain obvious shortcomings in the other measures. In contrast, the LPIPS similarity shows near-top measures all around with the most balanced performance, which is employed in our full method. For augmentation, it consistently benefits all the measures in general for all the distance metrics, which is also employed. In summary, harmonizing GAN and IMLE is a non-trivial challenge. Our two strategies achieve the best of the two worlds by significantly improving the overall performance (including data coverage) from the vanilla version.

For completeness, in Table 1 second part we also compare to VAEGAN which is alternatively incorporated with different distance metrics. Although LPIPS metric boosts our method the most, we find Inception space boosts VAEGAN the most. But it is still not as advantageous as our performance in general, especially for Recall and IvOM which corresponds to data coverage.

The radar plots in Figure 1 assist interpret Table 1.

Table 1: Comparisons on CelebA dataset. We indicate for each metric whether a higher (⇑) or lower (⇓) value is more desirable. We highlight the best performance in **bold** and the second best performance with underline. We visualize the radar plots in Figure 1 for the comprehensive evaluation of each method over the validation set.

	FID30k ⇓		Precision30k ⇑		Recall30k ⇑		IvOM3k ⇓		IvOM3k std ⇓	
Method	Train	Val	Train	Val	Train	Val	Train	Val	Train	Val
StyleGAN2 [3]	9.37	9.49	0.855	0.844	0.730	0.741	0.303	0.302	0.0268	0.0264
VAEGAN [5]	18.26	18.14	0.738	0.733	0.782	0.779	0.310	0.307	0.0264	0.0246
VAEGAN pixel	28.89	28.49	0.689	0.683	0.573	0.594	0.323	0.320	0.0259	0.0256
VAEGAN Inception	**8.35**	**8.47**	0.875	0.872	0.687	0.687	0.298	0.295	0.0248	0.0235
VAEGAN LPIPS	24.10	23.47	0.878	0.851	0.572	0.560	0.318	0.315	0.0284	0.0272
Ours pixel	34.94	34.46	0.774	0.771	0.751	0.763	0.272	0.280	0.0199	0.0222
Ours pixel interp	32.54	31.82	0.828	0.828	**0.882**	**0.879**	0.265	0.277	0.0207	0.0231
Ours Dfeature [5]	28.85	28.34	0.793	0.808	0.811	0.814	**0.255**	0.271	**0.0188**	0.0227
Ours Dfeature interp	22.38	21.92	0.849	0.842	0.806	0.826	0.263	0.277	0.0189	0.0219
Ours Inception [12]	14.86	14.95	0.859	0.853	0.675	0.706	0.294	0.299	0.0232	0.0237
Ours Inception interp	11.62	11.61	0.843	0.861	0.704	0.712	0.301	0.303	0.0234	0.0249
Ours LPIPS [16]	12.30	12.10	0.916	0.936	0.835	0.843	0.256	0.263	0.0194	**0.0195**
Ours LPIPS interp	11.56	11.28	**0.927**	**0.941**	0.849	0.848	**0.255**	**0.262**	0.0193	**0.0195**

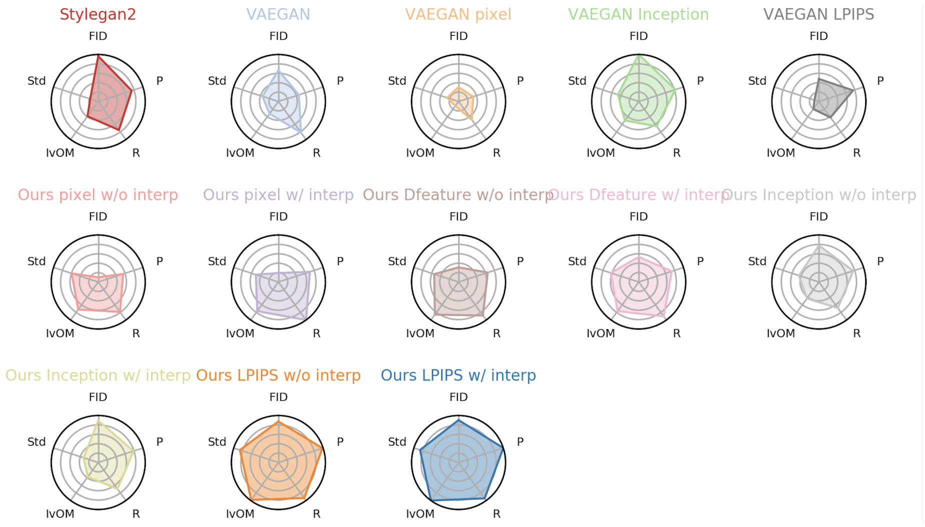


Fig. 1: Radar plots for Table 1. "P" represents Precision, "R" represents Recall, and "Std" represents IvOM standard deviation. Values have been normalized to the unit range, and axes are inverted so that the higher value is always better.

3 Additional Results on Minority Inclusion

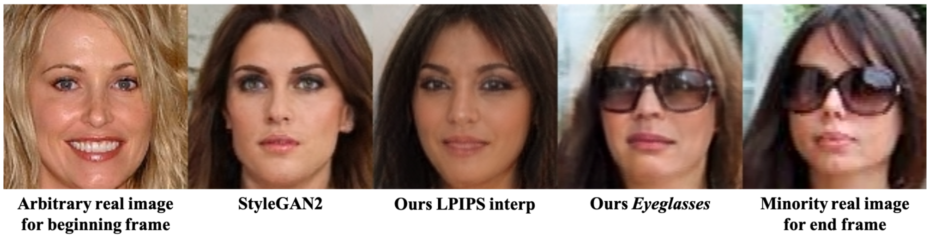
In order to dynamically demonstrate the effectiveness of our minority inclusion models, we are attaching four videos along with this supplementary material and also at GitHub. The videos show the results of interpolating in the latent space from one arbitrary image to another image with specific attribute(s). In this way we show our minority inclusion model variants perform comparably to the other models for majority groups, and outperform the others for minority groups.

In each video, the leftmost column is an arbitrary real image and the rightmost column is an arbitrary real image with specific attribute(s) of interest. For each generative model, we project the image in the leftmost column onto its latent space (i.e.: we find the latent vector that results in a generated image that is most perceptually similar to the image according to LPIPS [16]), and then interpolate starting from this latent vector. We do the same for the image in the rightmost column and use the resulting latent vector as the target for interpolation. The sub-videos in the middle three columns are the images produced by three methods: StyleGAN2 [3], our general IMLE-GAN model described in Section 3.3 and 4.4 in the main paper ("Ours LPIPS interp"), and our IMLE-GAN model with specific minority inclusion described in Section 3.4 and 4.5 in the main paper (Ours *attributeA*&*attributeB*). The four videos correspond to the four arbitrarily selected attributes or attribute combinations used in Section 4.5 in the main paper: *Eyeglasses*, *Bald*, *Narrow_Eyes*&*Heavy_Makeup* (*NE*&*HM*), and *Bags_Under_Eyes*&*High_Cheekbone*&*Attractive* (*BUE*&*HC*&*A*). For convenience, we show the last frame of each video in Figure 2, where each generated image is the projection of the rightmost image (a real image from the minority group) onto the space of images learned by each generative model.

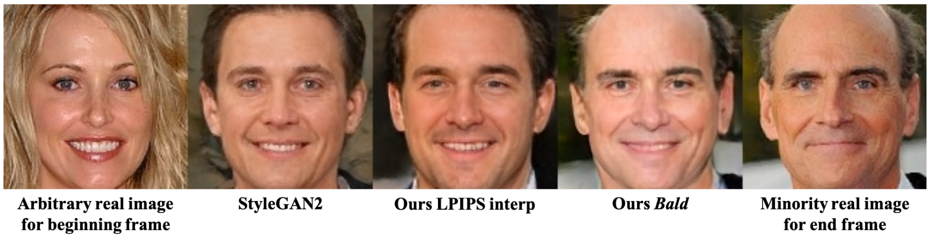
We note from the qualitative comparisons that incorporating minority inclusion in the training objective ensures coverage of the specified minority group, with little or no compromise from their performance on the majority. For example, in each video, at the beginning the three models are comparably representative for the arbitrary real image from the majority group (the leftmost column). As the latent vector transitions towards the corresponding minority region (the rightmost column), the attribute appearances of the minority group are not reconstructed accurately by the two models without an explicit focus on minority attributes (the second and third columns from the left). On the contrary, our minority inclusion model (the second column from the right) effectively represents the desired minority attributes, e.g., sunglasses, narrow eye shapes, or eye bags.

References

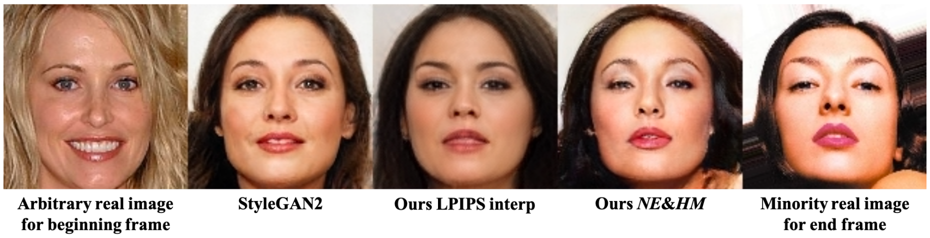
1. Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., Courville, A.: Adversarially learned inference. In: ICLR (2016) 1, 2
2. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017) 1, 2


Arbitrary real image for beginning frame
StyleGAN2
Ours LPIPS interp
Ours *Eyeglasses*
Minority real image for end frame

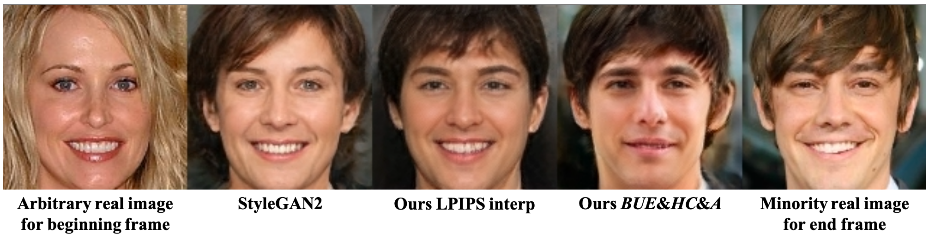
(a) Minority: *Eyeglasses*


Arbitrary real image for beginning frame
StyleGAN2
Ours LPIPS interp
Ours *Bald*
Minority real image for end frame

(b) Minority: *Bald*


Arbitrary real image for beginning frame
StyleGAN2
Ours LPIPS interp
Ours *NE&HM*
Minority real image for end frame

(c) Minority: *Narrow_Eyes&Heavy_Makeup*


Arbitrary real image for beginning frame
StyleGAN2
Ours LPIPS interp
Ours *BUE&HC&A*
Minority real image for end frame

(d) Minority: *Bags_Under_Eyes&High_Cheekbone&Attractive*

Fig. 2: The last frame of each video in the attachment and also at GitHub. Each of the middle three columns denotes a generated image from a learned model, the latent vector of which is projected from the image in the rightmost column (a real image from one minority subgroup).

3. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. arXiv (2019) 1, 3, 4
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2015) 1, 2
5. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: ICML (2016) 1, 2, 3
6. Lin, Z., Khetan, A., Fanti, G., Oh, S.: Pacgan: The power of two samples in generative adversarial networks. In: NeurIPS (2018) 2
7. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (2015) 1
8. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. In: ICLR (2017) 1, 2
9. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: ICLR (2018) 2
10. Rosca, M., Lakshminarayanan, B., Warde-Farley, D., Mohamed, S.: Variational approaches for auto-encoding generative adversarial networks. arXiv (2017) 1, 2
11. Sajjadi, M.S., Bachem, O., Lucic, M., Bousquet, O., Gelly, S.: Assessing generative models via precision and recall. In: NeurIPS (2018) 2
12. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NeurIPS (2016) 3
13. Srivastava, A., Valkov, L., Russell, C., Gutmann, M.U., Sutton, C.: Veegan: Reducing mode collapse in gans using implicit variational learning. In: NeurIPS (2017) 1, 2
14. Tran, N.T., Bui, T.A., Cheung, N.M.: Dist-gan: An improved gan using distance constraints. In: ECCV (2018) 1
15. Yang, D., Hong, S., Jang, Y., Zhao, T., Lee, H.: Diversity-sensitive conditional generative adversarial networks. In: ICLR (2019) 1
16. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 1, 2, 3, 4