Supplementary for Dual Projection Generative Adversarial Networks for Conditional Image Generation

Ligong Han,[1] Martin Renqiang Min,[2] Anastasis Stathopoulos,[1]
Yu Tian,[1] Ruijiang Gao,[3] Asim Kadav,[2] Dimitris Metaxas[1]
[1]Department of Computer Science, Rutgers University [2]NEC Labs America
[3]McCombs School of Business, The University of Texas at Austin
lh599@rutgers.edu, renqiang@nec-labs.com, as2947@cs.rutgers.edu
yt219@cs.rutgers.edu, ruijiang@utexas.edu, asim@nec-labs.com, dnm@cs.rutgers.edu

1. Proof of Proposition 1

Proposition 1. *When $\psi = 0$, a Proj-GAN reduces to K unconditional GANs, each of them minimizes the Jensen-Shannon divergence between $P_{X|y}$ and $Q_{X|y}$ with mixing ratio $\{\frac{P(y)}{P(y)+Q(y)}, \frac{Q(y)}{P(y)+Q(y)}\}$. Its value function can be written as,*

$$\mathbb{E}_{P_Y}\left\{\mathbb{E}_{P_{X|Y}}\log D(x|y) + \frac{Q_y}{P_y}\mathbb{E}_{Q_{X|Y}}\log\left(1 - D(x|y)\right)\right\}.$$

Proof. When $\psi(\cdot)$ is zero, $\tilde{D}(x, y) = v_y^{\mathrm{T}}\phi(x)$. Recall the logit of an unconditional GAN is $\tilde{D}(x) = v_X^{\mathrm{T}}\phi(x)$ (with bias $b = 0$). It immediately follows that matrix V is a collection of K vectors v_y, one for each class. Simply rearranging the cGAN objective, we get

$$\mathbb{E}_{P(y)}\left\{\mathbb{E}_{P(x|y)}\log D(x|y) + \frac{1}{r(y)}\mathbb{E}_{Q(x|y)}\log\left(1 - D(x|y)\right)\right\}, \tag{1}$$

with $r(y) = \frac{P(y)}{Q(y)}$. This can be viewed as a weighted sum of K GAN objectives with binary cross-entropy loss. Each of them minimizes the Jensen-Shannon divergence between $P_{X|y}$ and $Q_{X|y}$ with weights $\{\frac{P(y)}{P(y)+Q(y)}, \frac{Q(y)}{P(y)+Q(y)}\}$. □

2. Proof of Proposition 2

Lemma 1. *For any classifier C, the objective $\mathbb{E}_{x,y\sim P_{XY}}\log C(x, y) \leq -H_P(Y|X)$, and the maximizer C^* is obtained if and only if $Q^c(y|x) = P(y|x)$, where Q^c is the conditional distribution induced by C.*

Proof. It follows immediately with the observation,

$$\begin{aligned}
L_{\mathrm{CE}} =& \mathbb{E}_{x,y\sim P_{XY}}\log C(x, y)\\
=& \mathbb{E}_{x\sim P_X}\mathbb{E}_{y\sim P_{Y|X}}\log P(y|x)\frac{Q^c(y|x)}{P(y|x)}\\
=& \mathbb{E}_{x\sim P_X}\mathbb{E}_{y\sim P_{Y|X}}\log P(y|x) -\\
& \mathbb{E}_{x\sim P_X}\mathrm{KL}(P_{Y|X}\|Q^c_{Y|X})\\
\leq& - H_P(Y|X).
\end{aligned} \tag{2}$$

The equality is achieved if and only if $Q^c_{Y|X} = P_{Y|X}$. □

Proposition 2. *Given a generator G, if cross entropy losses L^p_{mi} and L^q_{mi} are minimized optimally, then the difference of two losses evaluated at fake data equals the reverse KL-divergence between $P_{Y|X}$ and $Q_{Y|X}$,*

$$L^p_{mi}(x^-) - L^q_{mi}(x^-) = \mathbb{E}_{Q_X}KL(Q_{Y|X}\|P_{Y|X}). \tag{3}$$

Proof. Applying Lemma 1 to classifier C^p and C^q respectively, we have

$$C^{p*} = P(y|x) \quad \text{and} \quad C^{q*} = Q(y|x). \tag{4}$$

Then,

$$\begin{aligned}
& L^{p*}_{mi}(x^-) - L^{q*}_{mi}(x^-)\\
=& \mathbb{E}_{z\sim P_Z, y\sim P_Y}\log Q^{cq*}(G(z, y), y) - \log Q^{cp*}(G(z, y), y)\\
=& \mathbb{E}_{z\sim P_Z, y\sim P_Y}\log\frac{C^{q*}(G(z, y), y)}{C^{p*}(G(z, y), y)}\\
=& \mathbb{E}_{Q_X}\log\frac{Q(y|x)}{P(y|x)}\\
=& \mathbb{E}_{Q_X}\mathrm{KL}(Q_{Y|X}\|P_{Y|X}).
\end{aligned}$$

□

3. Proof of Theorem 1

Theorem 1. *Denoting P_{XY} and Q_{XY} as the data distribution and the distribution induced by G, their Jensen-Shannon divergence is upper bounded by the following,*

$$JSD(P_{XY}, Q_{XY}) \leq \tag{5}$$
$$2c_1\sqrt{2JSD(P_X, Q_X)} + c_2\sqrt{2KL(P_{Y|X}\|Q^p_{Y|X})} +$$
$$c_2\sqrt{2KL(Q_{Y|X}\|Q^q_{Y|X})} + c_2\sqrt{2KL(Q^q_{Y|X}\|Q^p_{Y|X})}.$$

Proof. According to the triangle inequality of the total variation distance (TV, denoted as δ), we have

$$\begin{aligned}&\delta(P_{XY}, Q_{XY})\\ \leq&\underbrace{\delta(P_{XY}, P_{Y|X}Q_X)}_{①} + \underbrace{\delta(P_{Y|X}Q_X, Q_{XY})}_{②}.\end{aligned} \tag{6}$$

We can relax term ① using the definition of TV,

$$\begin{aligned}&\delta(P_{XY}, P_{Y|X}Q_X) = \delta(P_{Y|X}P_X, P_{Y|X}Q_X)\\ =&\frac{1}{2}\int\{|P_{Y|X}(y|x)P_X(x) - P_{Y|X}(y|x)Q_X(x)|\mu(x,y)\}\\ \overset{(a)}{\leq}&\frac{1}{2}\int|P_{Y|X}(y|x)|\mu(x,y)\int|P_X(x) - Q_X(x)|\mu(x,y)\\ \leq&c_1\delta(P_X, Q_X),\end{aligned} \tag{7}$$

where μ is a (σ-finite) measure, c_1 is an upper bound of $\int|P_{Y|X}(y|x)|\mu(x,y)$. (a) follows from the Hölder inequality. Similarly, for ② we have,

$$\begin{aligned}\delta(P_{Y|X}Q_X, Q_{XY}) =&\delta(P_{Y|X}Q_X, Q_{Y|X}Q_X)\\ \leq&c_2\delta(P_{Y|X}, Q_{Y|X}),\end{aligned} \tag{8}$$

and c_2 is an upper bound of $\int|Q_X(x)|\mu(x)$.
Then, using the triangle inequality of TV again,

$$\begin{aligned}&\delta(P_{Y|X}, Q_{Y|X})\\ \leq&\delta(P_{Y|X}, Q^p_{Y|X}) + \delta(Q^p_{Y|X}, Q^q_{Y|X}) + \delta(Q^q_{Y|X}, Q_{Y|X}).\end{aligned} \tag{9}$$

Combining Equation 6, 7, 8 and 9,

$$\begin{aligned}&\delta(P_{XY}, Q_{XY})\\ \leq&c_1\delta(P_X, Q_X) + c_2\delta(P_{Y|X}, Q_{Y|X})\\ \leq&\underbrace{c_1\delta(P_X, Q_X)}_{③} + \underbrace{c_2\delta(Q^p_{Y|X}, Q^q_{Y|X})}_{④} +\\ &\underbrace{c_2\delta(P_{Y|X}, Q^p_{Y|X}) + c_2\delta(Q^q_{Y|X}, Q_{Y|X})}_{⑤}.\end{aligned} \tag{10}$$

From above, we see that ③ is enforced by the unconditional GAN, ④ is minimized by the f-divergence term, and ⑤ is bounded by L^p_{mi} and L^q_{mi}.

Finally, using to Pinsker inequality [8] $\delta(P,Q) \leq \sqrt{\frac{1}{2}KL(P\|Q)}$, and Lemma 3 in [7] $\frac{1}{2}\delta^2(P,Q) \leq JSD(P,Q) \leq 2\delta(P,Q)$, we have,

$$\begin{aligned}&\text{JSD}(P_{XY}, Q_{XY}) \leq\\ &2c_1\sqrt{2\text{JSD}(P_X, Q_X)} + c_2\sqrt{2\text{KL}(Q^q_{Y|X}\|Q^p_{Y|X})} +\\ &c_2\sqrt{2\text{KL}(P_{Y|X}\|Q^p_{Y|X})} + c_2\sqrt{2\text{KL}(Q_{Y|X}\|Q^q_{Y|X})}.\end{aligned} \tag{11}$$

□

4. Weighted Dual Projection GAN

P2GAN-ap. The full objectives of P2GAN with amortised weights are as follows,

$$\begin{aligned}L_D^{P2ap} =&\mathbb{E}_{x,y\sim P_{XY}}(1-\lambda(x))\mathcal{A}(-\tilde{D}(x,y)) +\\ &\mathbb{E}_{z\sim P_Z, y\sim Q_Y}(1-\lambda(G(z,y)))\mathcal{A}(\tilde{D}(G(z,y),y)) -\\ &\mathbb{E}_{x,y\sim P_{XY}}\lambda(x)T^p(x,y) -\\ &\mathbb{E}_{z\sim P_Z, y\sim Q_Y}\lambda(G(z,y))T^q(G(z,y),y) \quad \text{and}\\ L_G^{P2ap} =&\mathbb{E}_{z\sim P_Z, y\sim Q_Y}(1-\lambda(G(z,y)))\mathcal{A}(-\tilde{D}(G(z,y),y)).\end{aligned} \tag{12}$$

Here T^p and T^q has the same definition as in f-cGAN.
Alternative weighing strategies. An alternative design of a weighted P2GAN is to fix the weight of L_D to 1,

$$L_D^{P2sp\text{-}alt} = L_D + \lambda\cdot(L^p_{mi} + L^q_{mi}) - \frac{1}{2}\log\lambda. \tag{13}$$

Here, $\lambda \in [0,\infty)$ and is initialized as 1. We can define similar alternatives for P2GAN-s, P2GAN-a and P2GAN-ap. The key difference is that, weighing $(1-\lambda)\cdot L_D$ and $\lambda\cdot L_{mi}$ explicitly balances data matching and label matching, while the alternative way balances L_{mi} and the penalty term. Without penalty, λ in all alternative variants will vanish since this minimizes the total loss, however, the decreasing rate is determined by loss L_{mi} adaptively. An extended comparison is listed in Table 1. In practice, we find the differences are not significant.

5. f-divergence

Here we consider several f-divergence loss functions [4] and list them in Table 2. Results on CIFAR100IB and VGGFace200 is given in Figure 1. Different from the results on VGGFace200, only reverse-KL and GAN losses are stable on CIFAR100IB.

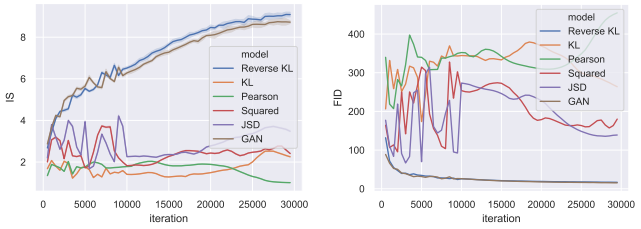
6. Implementation

The 1D Mixture of Gaussian experiments are implemented based on the official TAC-GAN repo*. Code for

*https://github.com/batmanlab/twin-auxiliary-classifiers-gan

Table 1: FID scores of alternative weighting strategies for P2GAN-w. All models are trained for 62000 iterations on CIFAR100 and 50000 iterations on VGGFace200.

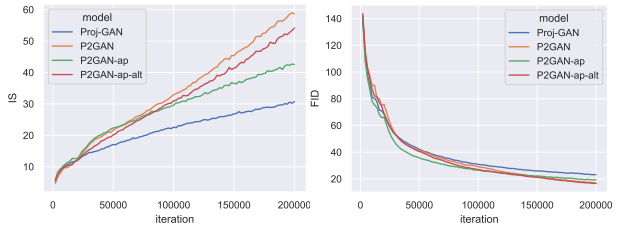
	CIFAR100	VGGFace200
P2GAN-s	**9.03**	20.59
P2GAN-sp	9.51	20.18
P2GAN-a	10.13	20.26
P2GAN-ap	9.82	**18.99**
P2GAN-s-alt	**9.49**	**20.21**
P2GAN-sp-alt	9.85	20.82
P2GAN-a-alt	9.98	23.25
P2GAN-ap-alt	9.72	21.57



(a) IS on CIFAR100IB (b) FID on CIFAR100IB

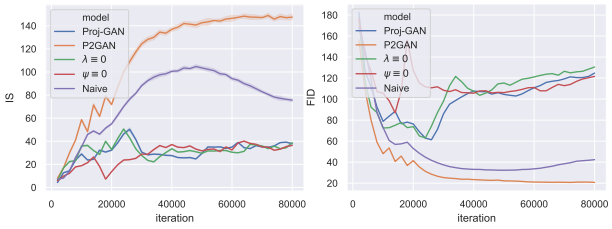
(c) IS on VGGFace200 (d) FID on VGGFace200

Figure 1: (a-b) IS and FID on CIFAR100IB. (c-d) IS and FID on VGGFace200. Different curves correspond to different choices of f-divergence, CE loss is used for both P and Q.



(a) IS on ImageNet (b) FID on ImageNet

Figure 2: IS and FID score over iterations on ImageNet at at 128×128 resolution.



(a) IS on VGGFace200 (b) FID on VGGFace200

Figure 3: IS and FID score over iterations on VGGFace200.

CIFAR100, VGGFace2, and ImageNet at resolution 64×64 are written based on the BigGAN-PyTorch repo[†]. Code for CIFAR10 and ImageNet at resolution 128×128 is implemented based on StudioGAN [2] repo[‡].

7. 1D MoG Synthetic Data

Experimental setup. We follow the same protocol as in TAC-GAN paper [1]. The standard deviations $\sigma_0 = 1$, $\sigma_1 = 2$, and $\sigma_2 = 3$ are fixed, and distance d_m is set to value $1, 2, \ldots, 5$ and all models are trained 100 times for each experimental setting. The code for synthetic data, network architectures, and MMD evaluation metrics are borrowed from the official TAC-GAN repo. However, the training code for hinge loss is not provided, thus we implemented our hinge loss version based on the BigGAN-PyTorch repo.

More results. The average MMD values across 100 runs are reported in Table 4 and Figure 4. Samples of generated 1D MoG are visualized in Figure 7. We observe that P2GAN performs the best with BCE loss, demonstrating its ability to generate accurate distributional data. Even with hinge loss, P2GAN still performs relatively well, and achieves the highest overall ranking.

8. CIFAR

Experimental setup. To construct the CIFAR100IB dataset, we randomly sample N_c images from class c where $N_c = \text{round}(500 - 4 \times c)$. For CIFAR100 experiments, we fix batch size as 100, and the number of D steps per G step as 4. All baselines are trained for 500 epochs or 62k iterations. These hyper-parameters are kept the same as described in TAC-GAN paper (also in their provided launch script).

More results. Generated samples of CIFAR10 and CIFAR100 are shown in Figure 8 and 10, respectively.

[†] https://github.com/ajbrock/BigGAN-PyTorch

[‡] https://github.com/POSTECH-CVLab/PyTorch-StudioGAN

Table 2: List of f-divergence and their corresponding *generator function* $f(\cdot)$

Name	$f(u)$	$f \circ \exp(u)$
Reverse KL	$-\log u$	$-u$
Kullback-leibler	$u \log u$	ue^u
Pearson χ^2	$(u-1)^2$	$(e^u-1)^2$
Squared Hellinger	$(\sqrt{u}-1)^2$	$(e^{u/2}-1)^2$
Jensen-Shannon	$-(u+1)\log\frac{1+u}{2} + u\log u$	$-(e^u+1)\log\frac{1+e^u}{2} + ue^u$
GAN	$u\log u - (u+1)\log(u+1)$	$ue^u - (e^u+1)\log(e^u+1)$

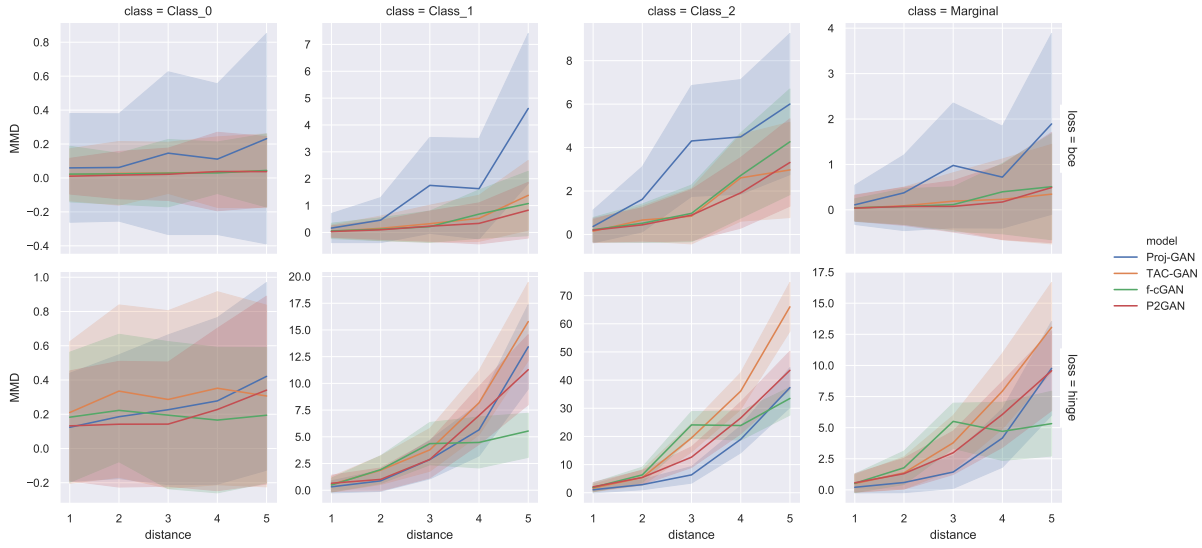


Figure 4: The Maximum Mean Discrepancy (MMD) metric. Proposed methods show low MMD with low variance across different runs.

Table 3: Inception Scores (IS), Fréchet Inception Distances (FID) and the maximum intra FID (max-FID), evaluated on VGGFace200 dataset.

	IS ↑	FID ↓	max-FID ↓
Proj-GAN	50.93 ± 0.86	61.43	239.61
TAC-GAN*	40.78 ± 0.57	96.06	478.10
Naïve	104.52 ± 1.95	32.39	196.99
f-cGAN	109.94 ± 1.15	29.54	215.50
P2GAN	148.48 ± 2.87	20.70	209.86
P2GAN-w	$\mathbf{171.31 \pm 3.44}$	**15.70**	**127.43**

9. ImageNet

Experimental setup. Due to limited computation resource, experiments on ImageNet related to model comparison and analysis are conducted at resolution 64×64. We follow the experimental setup in TAC-GAN paper but reduce the image size and model size. We use batch size of 2048 (batch size 256 accumulated 8 times) and the number of D steps per G step is 1. Channel multipliers for both G and D are 32. The resolution of self-attention layer is set to 32. Models are trained with 80k iterations.

For experiments at 128×128 resolution, we follow the configurations of BigGAN256[§] provided in the StudioGAN [2] repo. We use batch size of 256 and the number of D steps per G step is 2. Channel multipliers for both G and D are 96. The resolution of self-attention layer is set to 64. We train a P2GAN-w model with 200k iterations.

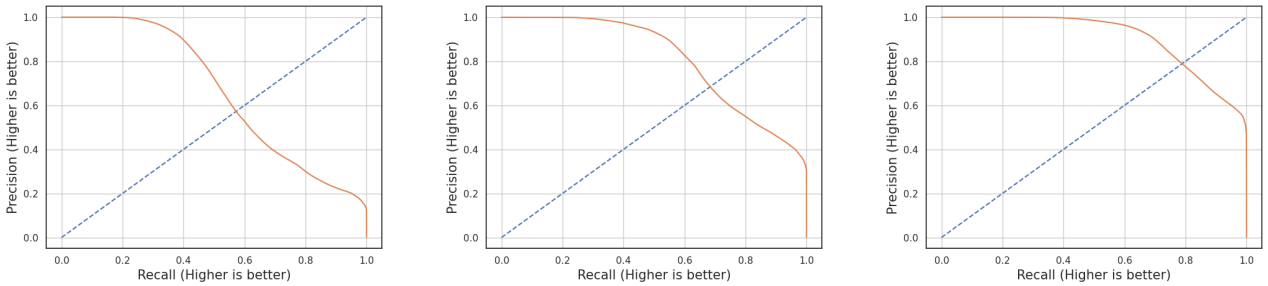
More results. Samples of 50 classes are shown in Figure 11 and 12. Although P2GAN (without adaptive weights) achieves the highest IS, it shows *mode collapse* on certain classes (for example the "flowers" in row 45). For the same flower class, Proj-GAN can still generate diverse samples. TAC-GAN, f-cGAN and P2GAN all exhibit mode collapse on certain classes. While the proposed weighting strategy is able to avoid mode collapse and still achieve competitively high IS and low FID.

Results of 128×128 resolution ImageNet experiments

[§] https://github.com/POSTECH-CVLab/PyTorch-StudioGAN/blob/master/src/configs/ILSVRC2012/BigGAN256.json

Table 4: The Maximum Mean Discrepancy (MMD) metric on 1D Mixture of Gaussian (MoG) synthetic dataset. Classes '0', '1', '2' stand for mode 0, 1, 2, and 'M' stands for marginal. The upper half lists results of BCE loss and the lower half lists results when adopting hinge loss. We run each experiment 100 times and report the average MMD over the top 90% performing runs. Standard deviations are omitted due to space limit. Entries with two lowest values are marked in boldface.

BCE / Hinge	$d_m = 1$				$d_m = 2$				$d_m = 3$				$d_m = 4$				$d_m = 5$			
	0	1	2	M	0	1	2	M	0	1	2	M	0	1	2	M	0	1	2	M
Proj-GAN	0.040	0.106	0.273	0.074	0.044	0.327	1.246	0.248	0.060	0.635	1.628	0.325	0.073	0.932	3.379	0.527	0.166	3.298	3.903	1.126
TAC-GAN*	**0.015**	**0.033**	**0.100**	**0.027**	0.021	0.124	0.529	0.067	**0.020**	0.272	0.803	0.149	**0.027**	**0.412**	**1.969**	**0.106**	**0.035**	1.139	**2.160**	**0.156**
f-cGAN	0.018	0.042	0.170	0.031	**0.019**	**0.090**	**0.383**	**0.047**	0.030	**0.193**	**0.635**	**0.087**	**0.024**	0.575	2.170	0.276	0.037	**0.857**	3.328	0.287
P2GAN	**0.009**	**0.028**	**0.151**	**0.026**	**0.014**	**0.080**	**0.345**	**0.046**	**0.016**	**0.160**	**0.639**	**0.084**	0.028	**0.237**	**1.530**	**0.056**	**0.030**	**0.655**	**2.725**	**0.261**
Proj-GAN	**0.112**	**0.267**	**0.879**	**0.178**	**0.167**	**0.725**	**2.373**	**0.492**	**0.172**	**1.455**	**6.385**	**0.969**	0.249	**4.904**	**15.496**	**3.368**	0.386	11.002	**29.382**	**7.407**
TAC-GAN*	0.190	0.474	**1.376**	**0.416**	0.304	1.635	**4.213**	1.060	0.357	3.504	**12.817**	**2.531**	0.314	6.949	29.822	6.425	**0.264**	13.905	54.134	11.125
f-cGAN	0.164	**0.429**	1.484	0.441	0.192	1.718	5.084	1.506	0.174	3.480	15.491	3.675	**0.138**	**3.629**	**19.597**	**3.862**	**0.173**	**4.592**	**28.753**	**4.315**
P2GAN	**0.118**	0.584	1.696	0.490	**0.120**	**0.843**	4.486	**1.051**	**0.152**	**2.951**	12.854	2.852	**0.192**	6.005	22.003	5.066	0.295	**9.920**	36.080	8.145



(a) Proj-GAN (b) P2GAN-w (c) P2GAN

Figure 5: Precision & recall on ImageNet (128 resolution).

are reported in Figure 2. The IS and FID curves over training iterations clearly shows its advantage in terms of fast convergence. Precision-recall [5] curves are given in Figure 5. Some randomly generated samples of P2GAN-w during training are shown in Figure 9.

10. VGGFace2

Experimental setup. We follow the same protocol as in TAC-GAN paper, and set batch size to 256 and the number of D steps per G step to 1. Images are resized to resolution of 64×64. The resolution of self-attention layer is set to 32. Channel multipliers for both G and D are 32. All baselines are trained with 100k iterations.

As for evaluation, we tried our best effort to match the calculated FID and IS with the reported values in TAC-GAN [1]. However, these values can be affected by many factors such as the selected subset of identities and the checkpoint of Inception Net [6] used for evaluation. We first sample a subset of 2000 identities and finetune an Inception model using Adam optimizer [3]. We use the checkpoinit at 20000 iteration to monitor the training of GAN models. Then we train a TAC-GAN model[¶] and select the best model with the lowest FID. Finally, we use the selected TAC-GAN model to examine which Inception Net checkpoint yields the best match. The final FID score is 29.54 which is very close to the reported 29.12. The identities of subsets VGGFace200, VGGFace500 and VGGFace2000 are given in Supplemental Materials.

More results. As a complementary to the t-SNE visualization of image embeddings provided in the main text, we visualize the samples and list the corresponding FID values in Figure 6. We see that Proj-GAN, TAC-GAN, f-cGAN and P2GAN all show *mode collapse* on identity 0 (the first row) while P2GAN-w still generates diverse samples on the given class. Additional IS and FID values are reported in Table 3.

The training curves of different baselines on VGGFace200 are plotted in Figure 3. We see that Proj-GAN, over-parameterzation baseline ($\lambda \equiv 0$), DM-GAN ($\psi \equiv 0$) and the naïve baseline all fail on VGGFace200. Samples of 50 identities from VGGFace500 are shown in Figure 13 and 14.

[¶]Here the model is chosen to be its actual implementation, which is equivalent to f-cGAN with reverse-KL and cross-entropy loss.

Figure 6: Samples and FID scores of VGGFace200, evaluated at iteration 2000, 20000, and 50000. Their identity numbers are 0, 1, and 2, respectively. At iteration 50000, all methods except for P2GAN-w exhibit mode collapse on identity 0.

References

[1] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin auxilary classifiers gan. In *Advances in Neural Information Processing Systems*, pages 1328–1337, 2019. 3, 5

[2] Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation. 2020. 3, 4

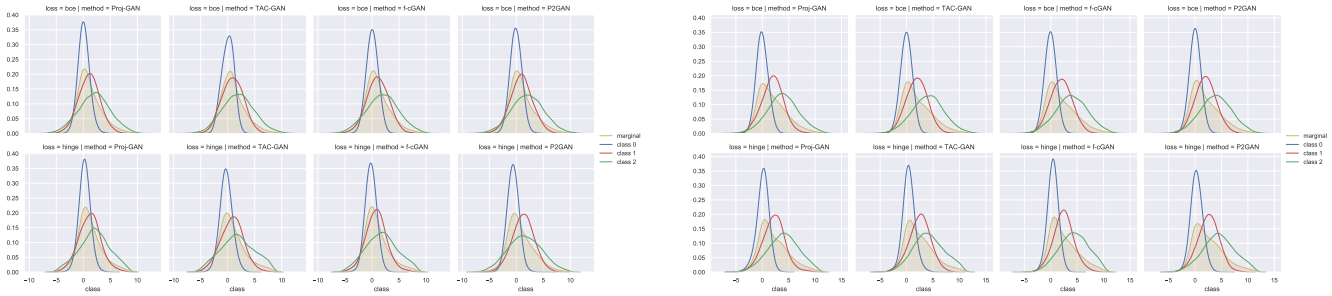
[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[4] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016. 2

[5] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*, 2018. 5

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 5

[7] Kiran Koshy Thekumparampil, Ashish Khetan, Zinan Lin,

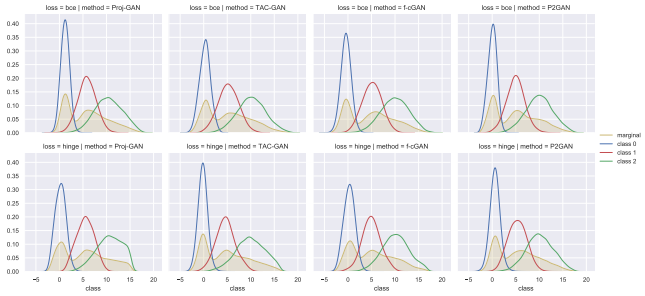


(a) distance $d_m = 1$

(b) distance $d_m = 2$

(c) distance $d_m = 3$

(d) distance $d_m = 4$



(e) distance $d_m = 5$

Figure 7: Change distance d_m between the means of adjacent 1-D Gaussian components. For each sub-figure, the first row adopts binary cross entropy loss and the second row adopts hinge loss.

and Sewoong Oh. Robustness of conditional gans to noisy labels. *arXiv preprint arXiv:1811.03205*, 2018. 2

[8] Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008. 2

(a) P2GAN (b) P2GAN-w

Figure 8: 10 classes of CIFAR10 generated samples at 32×32 resolution.

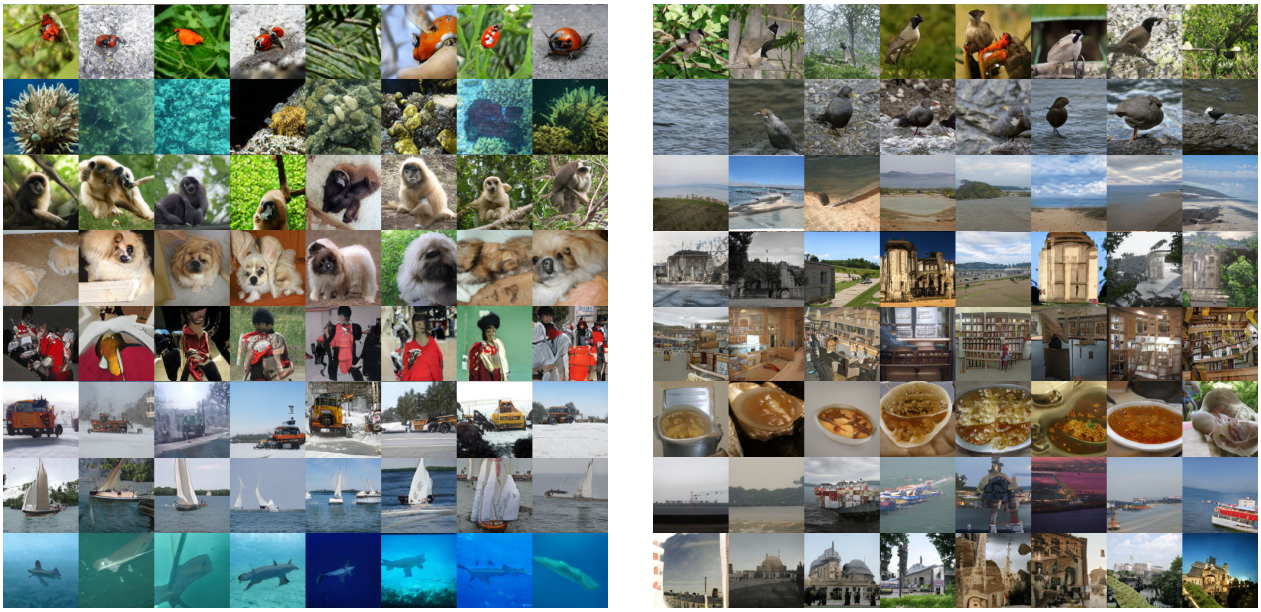
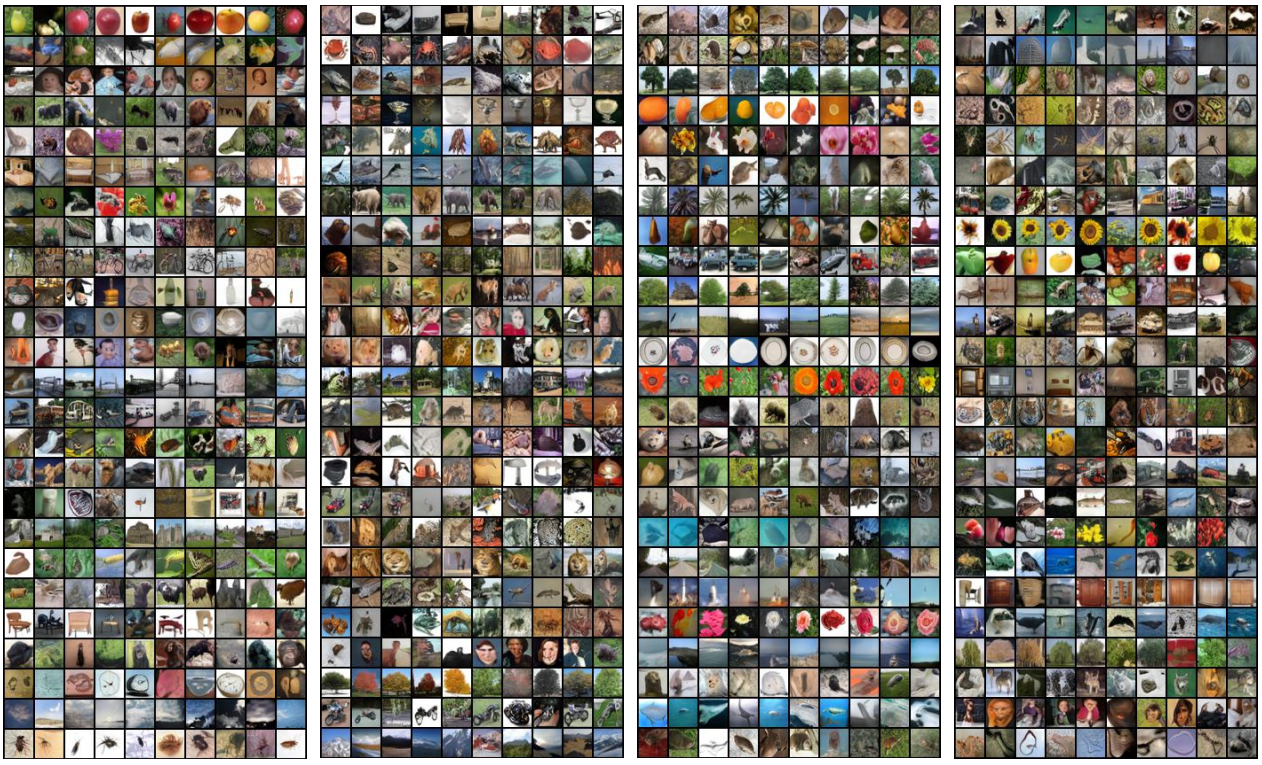


Figure 9: P2GAN-w generated samples on ImageNet at 128×128 resolution.

(a) P2GAN

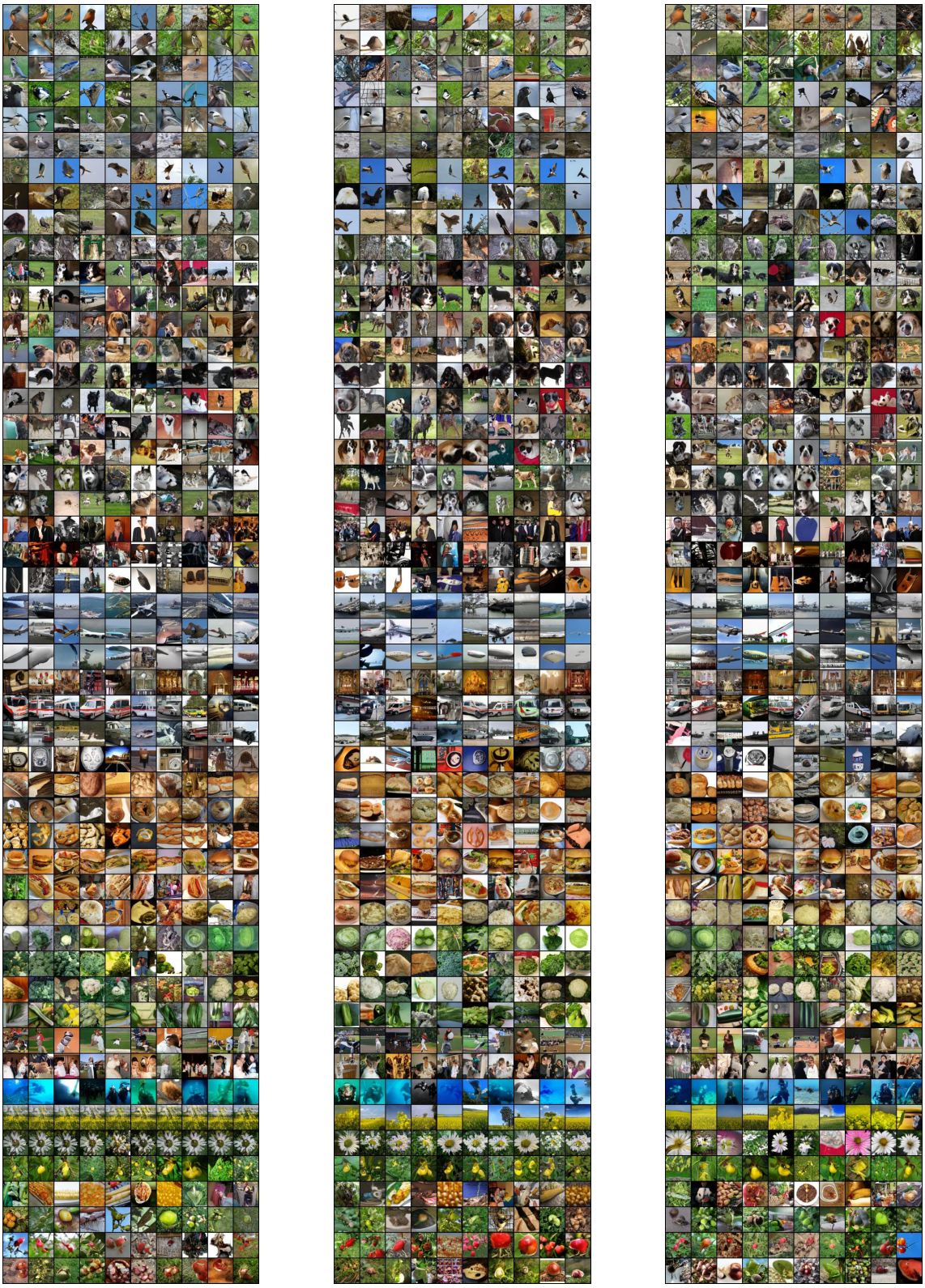


(b) P2GAN-w

Figure 10: 100 classes of CIFAR100 generated samples at 32×32 resolution.

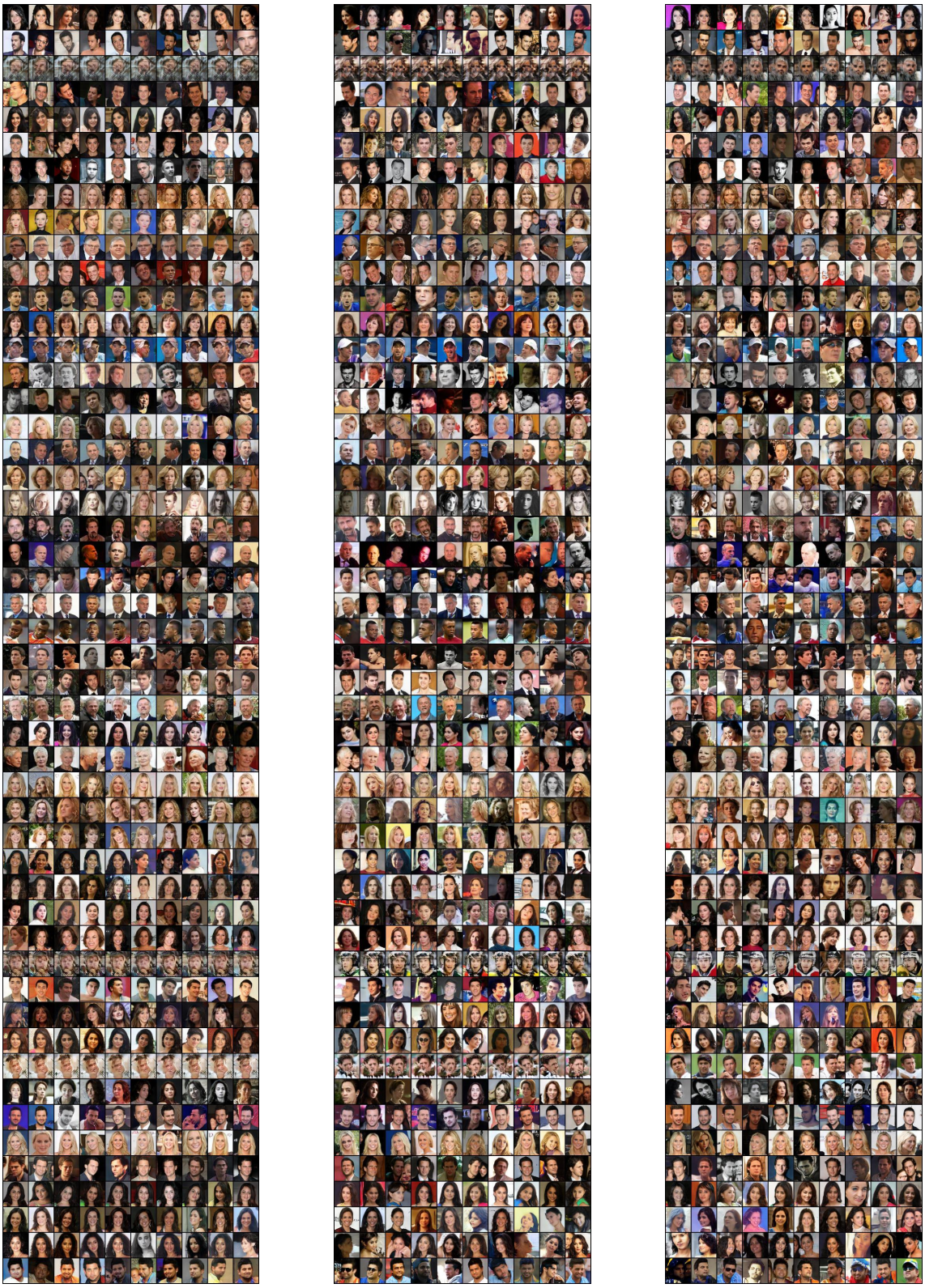
(a) Proj-GAN (b) TAC-GAN* (c) f-cGAN

Figure 11: 50 classes of ImageNet1000 generated samples at 64×64 resolution.



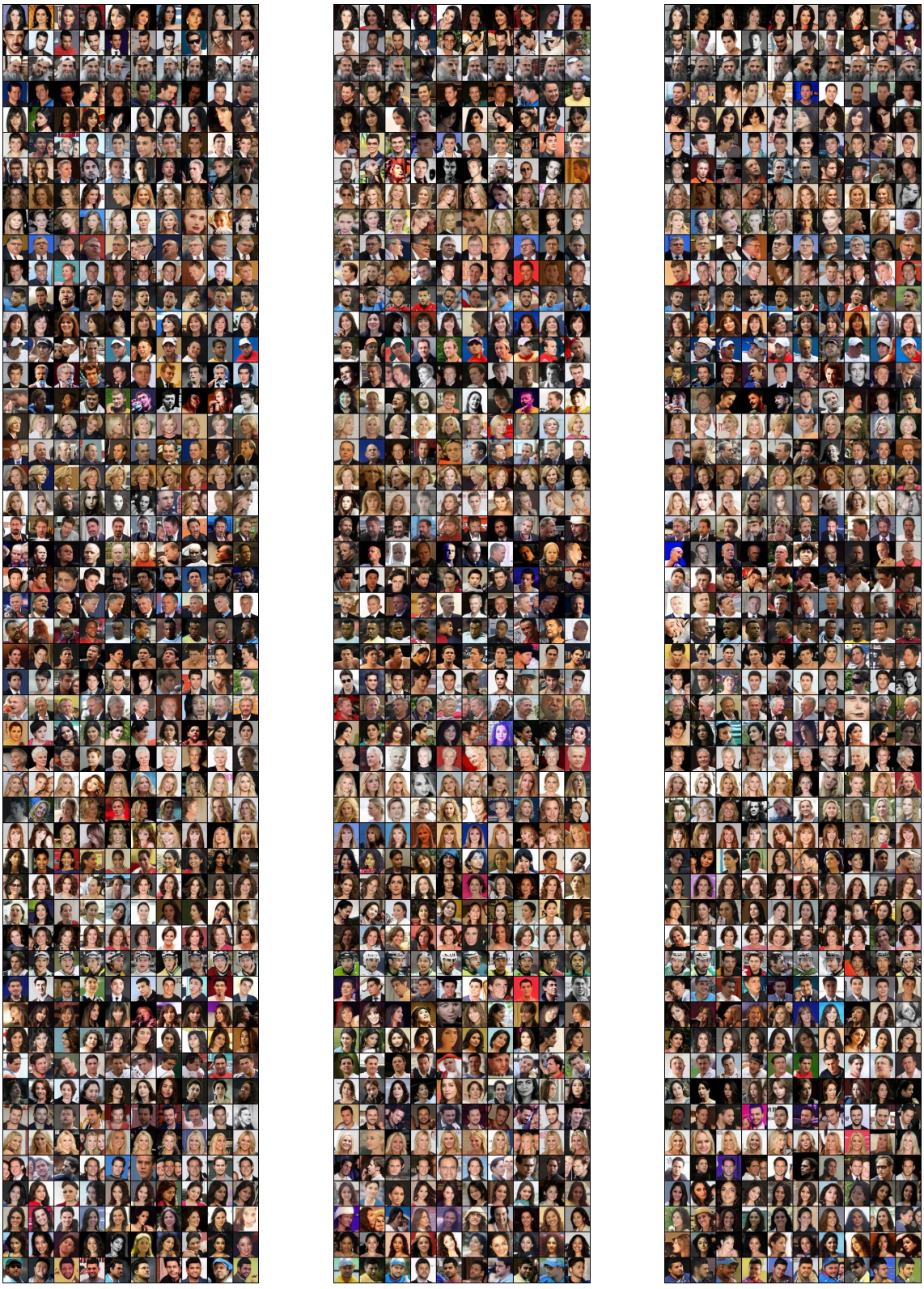
(a) P2GAN (b) P2GAN-a (c) P2GAN-ap

Figure 12: 50 classes of ImageNet1000 generated samples at 64×64 resolution.



(a) Proj-GAN (b) TAC-GAN* (c) f-cGAN

Figure 13: 50 classes of VGGFace500 generated samples at 64×64 resolution.



(a) P2GAN (b) P2GAN-a (c) P2GAN-ap

Figure 14: 50 classes of VGGFace500 generated samples at 64×64 resolution.