Supplementary for Dual Projection Generative Adversarial Networks for Conditional Image Generation

Ligong Han,1 Martin Renqiang Min,2 Anastasis Statopoulos,1 Yu Tian,1 Ruijiang Gao,3 Asim Kadav,2 Dimitris Metaxas1  
1Department of Computer Science, Rutgers University 2NEC Labs America  
3McCombs School of Business, The University of Texas at Austin  
lh599@rutgers.edu, renqiang@nec-labs.com, as2947@cs.rutgers.edu

1. Proof of Proposition 1

**Proposition 1.** When ψ = 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,

$$E_{P_{y}} \left\{ E_{P_{X|y}} \log D(x|y) + \frac{Q_{y}}{P_{y}} E_{Q_{X|y}} \log (1 - D(x|y)) \right\}.$$  

Proof. When ψ(·) is zero, $\hat{D}(x, y) = v_{y}^{T} \phi(x)$. Recall the logit of an unconditional GAN is $\hat{D}(x) = v_{X}^{T} \phi(x)$ (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

$$E_{P_{y}} \left\{ E_{P_{X|y}} \log D(x|y) + \frac{1}{r(y)} E_{Q_{X|y}} \log (1 - D(x|y)) \right\}.$$  

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 $E_{x,y \sim P_{X,Y}} \log C(x, y) \leq -H_{P}(Y|X)$, and the maximizer $C^{*}$ is obtained if and only if $Q^{*}(y|x) = P(y|x)$, where $Q^{*}$ is the conditional distribution induced by $C$.

**Proof.** It follows immediately with the observation,

$$L_{\text{CE}} = E_{x,y \sim P_{X,Y}} \log C(x, y)$$

$$= E_{x,y \sim P_{X,Y}} E_{P_{Y|X}} \log P(y|x) \frac{Q^{*}(y|x)}{P^{*}(y|x)}$$

$$= E_{x,y \sim P_{X,Y}} E_{P_{Y|X}} \log P(y|x) - E_{x \sim P_{X}} \text{KL}(P_{Y|X}||Q^{*}_{Y|X})$$

$$\leq -H_{P}(Y|X).$$  

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

**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 $Q_{Y|X}$.

$$L_{P}^{mi}(x^{-}) - L_{q}^{mi}(x^{-}) = E_{Q_{X}} \text{KL}(Q_{Y|X}||P_{Y|X}).$$  

Proof. Applying Lemma 1 to classifier $C^{P}$ and $C^{q}$ respectively, we have

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

Then,

$$L_{P}^{mi}(x^{-}) - L_{q}^{mi}(x^{-})$$

$$= E_{z \sim P_{z,Y}, y \sim P_{Y}} \log Q^{cq*}(G(z, y), y) - \log Q^{cq*}(G(z, y), y)$$

$$= E_{z \sim P_{z,Y}, y \sim P_{Y}} \log C^{cq*}(G(z, y), y)$$

$$= E_{Q_{X}} \log \frac{Q(y|x)}{P(y|x)}$$

$$= E_{Q_{X}} \text{KL}(Q_{Y|X}||P_{Y|X}).$$  

□
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.

\[
JSD(P_{XY}, Q_{XY}) \leq 2c_1 \sqrt{2JSD(P_X, Q_X)} + c_2 \sqrt{2KL(P_{Y|X} \| Q_{Y|X}^p)} + c_2 \sqrt{2KL(Q_{Y|X} \| Q_{Y|X}^p)}. \tag{5}
\]

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

\[
\delta(P_{XY}, Q_{XY}) \leq \delta(P_{XY}, P_{Y|X}Q_X) + \delta(P_{Y|X}Q_X, Q_{Y|X}Q_X). \tag{6}
\]

We can relax term (i) using the definition of TV,

\[
\delta(P_{XY}, P_{Y|X}Q_X) = \delta(P_{Y|X}P_X, P_{Y|X}Q_X) \leq \frac{1}{2} \int \left| P_{Y|X}(y|x)P_X(x) - P_{Y|X}(y|x)Q_X(x) \right| \mu(x, y) \tag{a}
\]

\[
\leq \frac{1}{2} \left( \int |P_{Y|X}(y|x)| \mu(x, y) \int |P_X(x) - Q_X(x)| \mu(x, y) \right) \leq c_1 \delta(P_X, Q_X), \tag{7}
\]

where $\mu$ is a ($\sigma$-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 (ii) we have,

\[
\delta(P_{Y|X}Q_X, Q_{Y|X}Q_X) \leq c_2 \delta(P_{Y|X}, Q_{Y|X}), \tag{8}
\]

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

\[
\delta(P_{Y|X}, Q_{Y|X}) \leq \delta(P_{Y|X}, Q_{Y|X}^p) + \delta(Q_{Y|X}^p, Q_{Y|X}) + \delta(Q_{Y|X}^p, Q_{Y|X}). \tag{9}
\]

Combining Equation 6, 7, 8 and 9,

\[
\delta(P_{XY}, Q_{XY}) \leq c_1 \delta(P_X, Q_X) + c_2 \delta(P_{Y|X}, Q_{Y|X}) \leq c_1 \delta(P_X, Q_X) + c_2 \delta(Q_{Y|X}^p, Q_{Y|X}) + c_2 \delta(P_{Y|X}, Q_{Y|X}^p) + c_2 \delta(Q_{Y|X}^p, Q_{Y|X}). \tag{10}
\]

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

\[
JSD(P_{XY}, Q_{XY}) \leq 2c_1 \sqrt{2JSD(P_X, Q_X)} + c_2 \sqrt{2KL(Q_{Y|X}^p \| Q_{Y|X})} + c_2 \sqrt{2KL(Q_{Y|X} \| Q_{Y|X}^p)}. \tag{11}
\]

\[\square\]

4. Weighted Dual Projection GAN

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

\[
L_D^{P2ap} = \mathbb{E}_{x, y \sim P_{XY}} (1 - \lambda(x))A(\tilde{D}(x, y)) + \mathbb{E}_{z \sim P_z, y \sim Q_Y} (1 - \lambda(G(z, y)))A(\tilde{D}(G(z, y), y)) - \mathbb{E}_{x, y \sim P_{XY}} \lambda(x)TP(x, y) - \mathbb{E}_{z \sim P_z, y \sim Q_Y} \lambda(G(z, y))T^q(G(z, y), y) \tag{12}
\]

Here $TP$ 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,

\[
L_D^{P2ap-alt} = L_D + \lambda \cdot (L_{mi}^p + L_{mi}^q) - \frac{1}{2} \log \lambda. \tag{13}
\]

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, $\lambda$ 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 \cite{4} and list them in Table 2. Results on CIFAR100IB and VGGFace200 is given in Table 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\footnote{https://github.com/batmanlab/twin-auxiliary-classifiers-gan}. Code for
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.

<table>
<thead>
<tr>
<th>Model</th>
<th>CIFAR100</th>
<th>VGGFace200</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2GAN-s</td>
<td>9.03</td>
<td>20.59</td>
</tr>
<tr>
<td>P2GAN-sp</td>
<td>9.51</td>
<td>20.18</td>
</tr>
<tr>
<td>P2GAN-a</td>
<td>10.13</td>
<td>20.26</td>
</tr>
<tr>
<td>P2GAN-ap</td>
<td>9.82</td>
<td><strong>18.99</strong></td>
</tr>
<tr>
<td>P2GAN-s-alt</td>
<td><strong>9.49</strong></td>
<td>20.21</td>
</tr>
<tr>
<td>P2GAN-sp-alt</td>
<td>9.85</td>
<td>20.82</td>
</tr>
<tr>
<td>P2GAN-a-alt</td>
<td>9.98</td>
<td>23.25</td>
</tr>
<tr>
<td>P2GAN-ap-alt</td>
<td>9.72</td>
<td>21.57</td>
</tr>
</tbody>
</table>

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 $Q$.

Figure 2: IS and FID score over iterations on ImageNet at 128 $\times$ 128 resolution.

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

CIFAR100, VGGFace2, and ImageNet at resolution 64 $\times$ 64 are written based on the BigGAN-PyTorch repo\(^1\). Code for CIFAR10 and ImageNet at resolution 128 $\times$ 128 is implemented based on StudioGAN [2] repo\(^2\).

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.

---

\(^1\)https://github.com/ajbrock/BigGAN-PyTorch
\(^2\)https://github.com/POSTECH-CVLab/PyTorch-StudioGAN
Table 2: List of $f$-divergence and their corresponding generator function $f(\cdot)$

<table>
<thead>
<tr>
<th>Name</th>
<th>$f(u)$</th>
<th>$f \circ \exp (u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse KL</td>
<td>$-\log u$</td>
<td>$-\exp u$</td>
</tr>
<tr>
<td>Kullback-Leibler</td>
<td>$u \log u$</td>
<td>$ue^u$</td>
</tr>
<tr>
<td>Pearson $\chi^2$</td>
<td>$(u-1)^2$</td>
<td>$(e^u-1)^2$</td>
</tr>
<tr>
<td>Squared Hellinger</td>
<td>$(\sqrt{u}-1)^2$</td>
<td>$(e^{u/2}-1)^2$</td>
</tr>
<tr>
<td>Jensen-Shannon</td>
<td>$-(u+1)\log \frac{1+u}{2} + u \log u$</td>
<td>$-(e^{u+1})\log \frac{1+e^u}{2} + ue^u$</td>
</tr>
<tr>
<td>GAN</td>
<td>$u \log u - (u+1)\log (u+1)$</td>
<td>$ue^u - (e^u+1)\log (e^u+1)$</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th></th>
<th>IS ↑</th>
<th>FID ↓</th>
<th>max-FID ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proj-GAN</td>
<td>50.93 ± 0.86</td>
<td>61.43</td>
<td>239.61</td>
</tr>
<tr>
<td>TAC-GAN*</td>
<td>40.78 ± 0.57</td>
<td>96.06</td>
<td>478.10</td>
</tr>
<tr>
<td>Naïve</td>
<td>104.52 ± 1.95</td>
<td>32.39</td>
<td>196.99</td>
</tr>
<tr>
<td>$f$-cGAN</td>
<td>109.94 ± 1.15</td>
<td>29.54</td>
<td>215.50</td>
</tr>
<tr>
<td>P2GAN</td>
<td>148.48 ± 2.87</td>
<td>20.70</td>
<td>209.86</td>
</tr>
<tr>
<td>P2GAN-w</td>
<td><strong>171.31 ± 3.44</strong></td>
<td><strong>15.70</strong></td>
<td><strong>127.43</strong></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>BCE / Hinge</th>
<th>$d_m = 1$</th>
<th>$d_m = 2$</th>
<th>$d_m = 3$</th>
<th>$d_m = 4$</th>
<th>$d_m = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proj-GAN</td>
<td>0.040</td>
<td>0.016</td>
<td>0.044</td>
<td>0.060</td>
<td>0.073</td>
</tr>
<tr>
<td>TAC-GAN*</td>
<td>0.015</td>
<td>0.003</td>
<td>0.019</td>
<td>0.030</td>
<td>0.010</td>
</tr>
<tr>
<td>f-cGAN</td>
<td>0.018</td>
<td>0.042</td>
<td>0.019</td>
<td>0.030</td>
<td>0.027</td>
</tr>
<tr>
<td>P2GAN</td>
<td>0.009</td>
<td>0.028</td>
<td>0.014</td>
<td>0.016</td>
<td>0.016</td>
</tr>
</tbody>
</table>

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

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 $64 \times 64$. 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 checkpoint at 20000 iteration to monitor the training of GAN models. Then we train a TAC-GAN model\(^3\) 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-parameterization 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.

---

\(^3\)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

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


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

Figure 9: P2GAN-w generated samples on ImageNet at 128 × 128 resolution.
Figure 10: 100 classes of CIFAR100 generated samples at $32 \times 32$ resolution.
Figure 11: 50 classes of ImageNet1000 generated samples at $64 \times 64$ resolution.
Figure 12: 50 classes of ImageNet1000 generated samples at $64 \times 64$ resolution.
Figure 13: 50 classes of VGGFace500 generated samples at 64 × 64 resolution.
Figure 14: 50 classes of VGGFace500 generated samples at $64 \times 64$ resolution.