Supplementary Material of
On Positive-Unlabeled Classification in GAN

Tianyu Guo[1,2], Chang Xu[2], Jiajun Huang[2], Yunhe Wang[3], Boxin Shi[4,5], Chao Xu[1], Dacheng Tao[2]
[1] Key Lab of Machine Perception (MOE), Dept. of Machine Intelligence, Peking University.
[2] UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering, University of Sydney.
[3] Noah's Ark Lab, Huawei Technologies. [4] NELVT, Dept. of CS, Peking University. [5] Peng Cheng Laboratory.
{tianyuguo, shiboxin}@pku.edu.cn, xuchao@cis.pku.edu.cn, yunhe.wang@huawei.com
{c.xu, dacheng.tao}@sydney.edu.au, jhua7177@uni.sydney.edu.au

Appendix A. Proofs

In this section, we provide proofs of theoretical results obtained in this paper.

A.1. Proof of Theorem 1

Theorem 1. *For the generator G fixed, the optimal discriminator D is*

$$D^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + \frac{1-\pi}{\pi} p_{gf}(\boldsymbol{x})},$$

where $p_g f(\boldsymbol{x})$ is the distribution of low-quality generated samples of G.

Proof. Given the generator G, the loss function can be written as

$$\begin{aligned}
\max_D V(D) &= \pi \int p_{data}(\boldsymbol{x}) \log(D(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} + \int p_z(\boldsymbol{z}) \log(1 - D(G(\boldsymbol{z})))\mathrm{d}\boldsymbol{z} - \pi \int p_{data}(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} \\
&= \pi \int p_{data}(\boldsymbol{x}) \log(D(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} + \int p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} - \pi \int p_{data}(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} \\
&= \int \pi p_{data}(\boldsymbol{x}) \log(D(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} + \int (1-\pi) p_{gf} \log(1 - D(\boldsymbol{x}))\mathrm{d}\boldsymbol{x}
\end{aligned} \tag{1}$$

.

Following the proof in GAN [1], the function $V(D)$ achieves its maximum at $\frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + \frac{1-\pi}{\pi} p_{gf}(\boldsymbol{x})}$. □

A.2. Proof of Theorem 2

Theorem 2. *With the optimal discriminator D fixed, the optimization of generator G is equivalent to minimize $\pi \log \pi + (1 - \pi) \log(1 - \pi) + \pi KL(p_{data}||p_g) + (1 - \pi) KL(p_{gf}||p_g)$.*

Proof. Theorem 1 proofs that the optimal discriminator $D^*_G(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + \frac{1-\pi}{\pi} p_{gf}(\boldsymbol{x})}$, then we can reformulate the minimal game of the generator G as following,

$$\begin{aligned}\min_G V(G) =&\pi \int p_{data}(\boldsymbol{x}) \log \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + \frac{1-\pi}{\pi} p_{gf}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \\ &+ \int p_g(\boldsymbol{x}) \log(1 - \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + \frac{1-\pi}{\pi} p_{gf}(\boldsymbol{x})}) \mathrm{d}\boldsymbol{x} - \pi \int p_{data}(\boldsymbol{x}) \log(1 - \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + \frac{1-\pi}{\pi} p_{gf}(\boldsymbol{x})}) \mathrm{d}\boldsymbol{x} \\ =&\pi \int p_{data}(\boldsymbol{x}) \log \frac{\pi p_{data}(\boldsymbol{x})}{\pi p_{data}(\boldsymbol{x}) + (1-\pi) p_{gf}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} + (1-\pi) \int p_{gf}(\boldsymbol{x}) \log \frac{(1-\pi) p_{gf}(\boldsymbol{x})}{\pi p_{data}(\boldsymbol{x}) + (1-\pi) p_{gf}(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \\ =&\pi \int p_{data}(\boldsymbol{x}) \log \frac{\pi p_{data}(\boldsymbol{x})}{p_g(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} + (1-\pi) \int p_{gf}(\boldsymbol{x}) \log \frac{(1-\pi) p_{gf}(\boldsymbol{x})}{p_g(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} \\ =&\pi \log \pi + (1-\pi) \log(1-\pi) + \pi KL(p_{data}||p_g) + (1-\pi) KL(p_{gf}||p_g)\end{aligned} \tag{2}$$

which completes the proof. □

Corollary 1. *The global minimum of the proposed objective function $V(G, D)$ is achieved if and only if $p_{gf} = p_g = p_{data}$. At that point, $C(G)$ achieves the value of $\pi \log \pi + (1-\pi) \log(1-\pi)$, and $D(\boldsymbol{x})$ achieves the value of π.*

Proof. Following the result of Theorem 2, we can easily conclude that the global minimum of $V(G)$ is $\pi \log \pi + (1-\pi) \log(1-\pi)$ if and only if $p_{gf} = p_g = p_{data}$, as the KL divergence between two distribution is non-negative and zero if they are equal.

At the same time the optimal $V(D)$ achieves $D^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + \frac{1-\pi}{\pi} p_{gf}(\boldsymbol{x})} = \frac{\pi p_{data}(\boldsymbol{x})}{\pi p_g(\boldsymbol{x}) + (1-\pi) p_{gf}(\boldsymbol{x})} = \pi$. □

Appendix B. Loss functions used in experiments

Here we show detailed loss functions of the proposed method integrated into various GAN frameworks. Flexibility to combining with the other excellent framework provides our approach with a chance to enhance generative performance based on the art-of-the-state results.

B.1. PUSGAN

$$\begin{aligned}\max_D V(D) =&\pi \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[\log(D(\boldsymbol{x}))] \\ &+ \max\{0, \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \\ &- \pi \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[\log(1 - D(\boldsymbol{x}))]\}.\end{aligned} \tag{3}$$

$$\min_G V(G) = \mathbb{E}_{\boldsymbol{z} \sim p_z}[\log(D(G(\boldsymbol{z})))] \tag{4}$$

B.2. PULSGAN (1 and -1 label)

$$\max_D V(D) = \pi \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[(D(\boldsymbol{x}) - 1)^2] + \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[(D(G(\boldsymbol{z})) + 1)^2] - \pi \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[(D(\boldsymbol{x}) + 1)^2]. \tag{5}$$

$$\min_G V(G) = \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[(D(G(\boldsymbol{z})) - 1)^2] \tag{6}$$

B.3. PUHingeGAN

$$\begin{aligned}\max_D V(D) =&\pi \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[\max(0, 1 - D(\boldsymbol{x}))] \\ &+ \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[\max(0, 1 + D(G(\boldsymbol{z})))] \\ &- \pi \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[\max(0, 1 + D(\boldsymbol{x}))]\}.\end{aligned} \tag{7}$$

$$\min_G V(G) = -\mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[D(G(z))] \tag{8}$$

B.4. PUSGAN-GP

$$\max_D V(D) = \pi \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[(D(x))] - \max\{0, \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[D(G(z))] - \pi \mathbb{E}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[D(x)]\} + \lambda \mathbb{E}_{\widehat{\boldsymbol{x}} \sim P_{\widehat{\boldsymbol{x}}}}[(|| \bigtriangledown_{\widehat{\boldsymbol{x}}} C(\widehat{\boldsymbol{x}})||_2 - 1)] \quad (9)$$

$$\min_G V(G) = \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[D(G(z))] \quad (10)$$

where $\mathbb{P}_{\widehat{\boldsymbol{x}}}$ is the distribution of $\widehat{\boldsymbol{x}} = \epsilon \boldsymbol{x}_r + (1-\epsilon)\boldsymbol{x}_f$, and $\boldsymbol{x}_r \sim p_{data}(\boldsymbol{x}), \boldsymbol{x}_f \sim p_g(\boldsymbol{x}), \epsilon \sim U[0,1]$.

Appendix C. Architectures

In this part, we display the architectures we used on various image datasets.

C.1. Structure for the 32×32 resolution

Generator
$z \in \mathbb{R}^{128} \sim N(0, I)$
Linear layer, $128 \to 512 \cdot 4 \cdot 4$
Reshape, $512 \cdot 4 \cdot 4 \to 512 \times 4 \times 4$
ConvTranspose2d: (512, 256, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d: (256, 128, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d: (128, 64, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d: (64, output channel, 3, 1, 1)
Tanh

Discriminator
$x \in \mathbb{R}^{3x32x32}$
Conv2d (3, 64, 3, 1, 1)
LeakyReLU 0.2
Conv2d (64, 64, 4, 2, 1)
LeakyReLU 0.1
Conv2d (64, 128, 3, 1, 1)
LeakyReLU 0.1
Conv2d (128, 256, 3, 1, 1)
LeakyReLU 0.1
Conv2d (256, 256, 4, 2, 1)
LeakyReLU 0.1
Conv2d (256, 512, 3, 1, 1)
Reshape, $512 \times 4 \times 4 \to 512 \cdot 4 \cdot 4$
Linear layer, $512 \cdot 4 \cdot 4 \to 1$

C.2. DCGAN for the 64×64 resolution

Generator
$z \in \mathbb{R}^{128} \sim N(0, I)$
ConvTranspose2d (128, 512, 4, 1, 0)
BatchNorm2d and ReLU
ConvTranspose2d (512, 256, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (256, 128, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (128, 64, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (64, output channel, 4, 2, 1)
Tanh

Discriminator
$x \in \mathbb{R}^{3x64x64}$
Conv2d (3, 64, 4, 2, 1)
LeakyReLU 0.2
Conv2d (64, 128, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (128, 256, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (256, 512, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (512, 1, 4, 2, 1)

C.3. Structure for the 128×128 resolution

Generator
$z \in \mathbb{R}^{128} \sim N(0, I)$
ConvTranspose2d (128, 1024, 4, 1, 0)
BatchNorm2d and ReLU
ConvTranspose2d (1024, 512, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (512, 256, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (256, 128, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (128, 64, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (64, output channel, 4, 2, 1)
Tanh

Discriminator
$x \in \mathbb{R}^{3x128x128}$
Conv2d (3, 64, 4, 2, 1)
LeakyReLU 0.2
Conv2d (64, 128, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (128, 256, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (256, 512, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (512, 1024, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (1024, 1, 4, 2, 1)

C.4. DCGAN for the 256×256 resolution

Generator
$z \in \mathbb{R}^{128} \sim N(0, I)$
ConvTranspose2d (128, 1024, 4, 1, 0)
BatchNorm2d and ReLU
ConvTranspose2d (1024, 1024, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (1024, 512, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (512, 256, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (256, 128, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (128, 64, 4, 2, 1)
BatchNorm2d and ReLU
ConvTranspose2d (64, output channel, 4, 2, 1)
Tanh

Discriminator (PACGAN2 [2])
$x \in \mathbb{R}^{3x256x256}$
Concatenate $[x_1, x_2] \in \mathbb{R}^{6x256x256}$
Conv2d (6, 32, 4, 2, 1)
LeakyReLU 0.2
Conv2d (32, 64, 4, 2, 1)
LeakyReLU 0.2
Conv2d (64, 128, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (128, 256, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (256, 512, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (512, 1024, 4, 2, 1)
BatchNorm2d and LeakyReLU 0.2
Conv2d (1024, 1, 4, 2, 1)

References

[1] 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. 1

[2] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In *Advances in neural information processing systems*, 2018. 4