

基于双向约束的生成对抗网络*

苟瑶¹, 李敏¹, 杜卫东^{1,2}, 何玉杰¹, 吴肇青¹, 宋雨^{1,3}



¹(火箭军工程大学 作战保障学院, 陕西 西安 710025)

²(武警工程大学 密码工程学院, 陕西 西安 710086)

³(国防科技大学 信息与通信学院, 湖北 武汉 430035)

通信作者: 李敏, E-mail: proflimin@163.com

摘要: 提高生成样本的质量和多样性一直是生成对抗网络 (generative adversarial network, GAN) 领域主要挑战任务之一。为此, 提出了一种双向约束生成对抗网络 (bidirectional constraint generative adversarial network, BCGAN)。与传统 GAN 变体相比, 该网络在架构设计上增加了一个生成器模块, 两个生成器分别从两个不同方向逼近真实样本的数据分布。然后根据 BCGAN 的网络架构, 设计了新的损失函数, 并对其进行了理论分析及证明。在 BCGAN 的训练过程中, 一方面通过增加两个生成样本数据分布之间的距离来丰富生成样本的多样性, 另一方面通过减小鉴别器对两个生成样本数据分布之间的差异来稳定训练过程, 提高生成样本的质量。最后, 在 1 个合成数据集和 3 个不同公开挑战数据集上进行了实验。一系列实验证明, 较其他生成方法相比, 所提方法对真实数据分布具有更强的拟合能力, 能够有效提升生成样本的质量和多样性。此外, 所提方法的训练过程更加平滑稳定。

关键词: 生成对抗网络; 双向约束; 样本多样性; 数据分布

中图法分类号: TP18

中文引用格式: 苟瑶, 李敏, 杜卫东, 何玉杰, 吴肇青, 宋雨. 基于双向约束的生成对抗网络. 软件学报, 2023, 34(9): 4195–4209. <http://www.jos.org.cn/1000-9825/6655.htm>

英文引用格式: Gou Y, Li M, Du WD, He YJ, Wu ZQ, Song Y. Generative Adversarial Network Based on Bidirectional Constraints. Ruan Jian Xue Bao/Journal of Software, 2023, 34(9): 4195–4209 (in Chinese). <http://www.jos.org.cn/1000-9825/6655.htm>

Generative Adversarial Network Based on Bidirectional Constraints

GOU Yao¹, LI Min¹, DU Wei-Dong^{1,2}, HE Yu-Jie¹, WU Zhao-Qing¹, SONG Yu^{1,3}

¹(College of Operational Support, Rocket Force University of Engineering, Xi'an 710025, China)

²(College of Cryptographic Engineering, Engineering University of PAP, Xi'an 710086, China)

³(College of Information and Communication, National University of Defense Technology, Wuhan 430035, China)

Abstract: Improving the quality and diversity of generated samples has always been one of the main challenging tasks in the field of generative adversarial network (GAN). For this reason, a bidirectional constraint GAN (BCGAN) is proposed. Compared with the traditional GAN variants, this network adds one more generator module to the architecture design. The two generators approach the data distribution of real samples from two different directions. Then, according to the network architecture of BCGAN, this study designs a new loss function and analyzes and proves it theoretically. During BCGAN training, the diversity of the generated samples is enriched by increasing the distance between the data distribution of two generated samples, and the difference of the discriminator between the data distribution of the two generated samples is reduced to stabilize the training process and thereby improve the quality of the generated samples. Finally, experiments are carried out on a synthetic dataset and three open challenge datasets. This series of experiments show that compared with other generative methods, the proposed method fits real data distribution better and effectively improves the quality and diversity of generated samples. In addition, the training process of this method is smoother and more stable.

Key words: generative adversarial network (GAN); bidirectional constraint; sample diversity; data distribution

* 基金项目: 国家自然科学基金 (62006240)

收稿时间: 2021-08-10; 修改时间: 2021-10-25, 2022-01-10; 采用时间: 2022-02-24; jos 在线出版时间: 2022-12-16

CNKI 网络首发时间: 2022-12-16

由于强大的生成能力, GAN^[1] 已经成为深度学习领域中最重要研究热点之一^[2]. GAN 通常包括一个生成器(generator, G) 和一个鉴别器(discriminator, D), 采用博弈论方法不断逼近生成样本与真实样本之间的数据分布. 和其他技术一样, GAN 也面临着一些挑战, 如生成样本质量不高, 生成样本往往缺乏多样性以及在训练过程中容易出现模型崩溃等^[3-5]. 为了缓解上述问题, GAN 变体主要从网络架构和损失函数两个方面进行了优化.

基于网络架构优化方面, 2015 年, Radford 等人^[6] 提出了一种深度卷积生成对抗网络(deep convolutional generative adversarial network, DCGAN), 首次引入卷积神经网络(convolutional neural network, CNN) 来设计 GAN 的生成器和鉴别器, 同时通过一些经验性的网络架构设计防止模型崩溃, 之后的大多数 GAN 变体模型都基于此体系结构进行改进的. 条件生成对抗网络(conditional generative adversarial network, CGAN)^[7] 则利用一些真实样本所包含的额外信息(如类别标签) 来控制生成样本内容. 而信息生成对抗网络(information generative adversarial network, infoGAN)^[8] 是在 CGAN 架构基础上开发的, 使得生成过程更加可控. 2021 年, 李响等人^[9] 通过多条件对抗和梯度优化方法, 一定程度上避免了梯度消失带来的影响, 进一步稳定了模型训练过程. 此外, 基于自编码器的发展也是优化 GAN 网络架构的一类方法^[10-14]. 然而, 上述 GAN 变体往往只能生成一些分辨率较低的图像, 如分辨率为 64×64 或 32×32 . 还有一些针对特定问题而设计的 GAN 模型, 如 BigGAN^[15]、StyleGAN^[16]、StyleGAN2^[17] 以及 ALIS^[18] 等, 这些 GAN 模型虽然能够在训练它们的任务上表现良好, 但由于其自身都是基于大规模卷积网络进行训练的, 从而大大增加了计算资源和训练时间, 限制了这些 GAN 模型的应用与发展.

基于损失函数优化方面, 文献[19] 研究表明, 任何散度都可以用于设计 GAN 的损失函数, 以稳定训练过程. 文献[20,21] 利用 Wasserstein 距离代替原来的 JS 散度, 一定程度上稳定了训练过程. 类似地, 文献[22-25] 都是利用不同散度来设计损失函数, 增强训练过程的稳定性. 2020 年, Xiangli 等人^[26] 提出了一种真实性生成对抗网络(realness generative adversarial network, RealnessGAN), 将真实性分布(realness distribution) 引入到 GAN 的训练中, 利用该方法, 即使在简单的 DCGAN 结构上, 也能够生成高分辨率的图像.

上述 GAN 变体利用不同方法稳定了训练过程, 然而这些 GAN 变体仅考虑从一个方向拟合真实样本的数据分布, 其生成样本的数据分布往往与真实样本的数据分布存在较大偏差. 为缓解这一问题, 同时进一步提高 GAN 生成样本的质量和多样性, 本文提出了一种双向约束生成对抗网络(bidirectional constraint generative adversarial network, BCGAN), 与传统 GAN 变体相比, BCGAN 在网络架构设计上增加了一个生成器模块, 两个生成器分别从两个不同方向逼近真实样本的数据分布. 根据 BCGAN 的网络架构, 设计了新的损失函数, 并对其进行了理论分析及证明. 然后, 本文探讨了损失函数的约束项对网络模型训练过程的影响, 通过在损失函数中增加约束项来优化 BCGAN 的性能, 主要措施有: (1) 通过增加两个生成样本数据分布之间的距离来丰富生成样本的多样性; (2) 通过减小鉴别器对两个生成样本数据分布之间的差异来稳定训练过程, 提高生成样本的质量.

本文第 1 节简要介绍了 GAN 以及 RealnessGAN 的基础知识. 第 2 节构建了 BCGAN 的网络架构和损失函数, 并对其损失函数的设计进行了理论分析及证明. 第 3 节是实验部分, 一系列实验表明本文方法生成样本的质量和多样性要优于 RealnessGAN, 且训练过程更加平稳. 第 4 节对全文进行了总结.

1 相关工作

1.1 GAN

GAN 通常由一个生成器 G 和一个鉴别器 D 组成, G 试图生成真伪难辨的样本, 而 D 则希望准确判别出样本的真伪. G 和 D 的训练过程遵循如下目标函数:

$$\begin{aligned} \min_G \max_D V(G, D) &= E_{x \sim p_{\text{data}}} [\log D(x)] + E_{z \sim p_z} [\log (1 - D(G(z)))] \\ &= E_{x \sim p_{\text{data}}} [\log D(x) - 0] + E_{x \sim p_g} [\log (1 - D(x))] \end{aligned} \quad (1)$$

其中, p_{data} 、 p_z 以及 p_g 分别表示真实样本、输入随机向量以及生成样本的数据分布.

文献[1] 已经证明, 当固定 G 时, 最优 D 满足 $D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$; 当固定 D 时, 最优 G 满足 $p_g = p_{\text{data}}$. 此时, 公式(1) 转化为如下形式:

$$\min_G \max_D V(G, D) = E_{x \sim p_{\text{data}}} \left[\log \frac{2p_{\text{data}}}{p_{\text{data}} + p_g} \right] + E_{x \sim p_g} \left[\log \frac{2p_g}{p_{\text{data}} + p_g} \right] - 2 \log 2 = 2JS(p_{\text{data}} \| p_g) - 2 \log 2 \quad (2)$$

因此, 可以通过最小化 p_{data} 和 p_g 之间的 JS 散度, 使得生成样本与真实样本足够相似. 然而, 如果 p_{data} 和 p_g 几乎没有重叠, $JS(p_{\text{data}} \| p_g)$ 将会是一个常数, 此时利用梯度下降法求解便会出现梯度消失, 这导致训练过程不稳定且不易收敛.

1.2 RealnessGAN

原始 GAN 认为 D 的输出是一个标量, 即鉴别输入样本 x 真伪的概率值 $D(x)$. RealnessGAN 对上述解释进行了推广, 认为 D 的输出是一个真实性分布向量 p_{realness} , 即鉴别输入样本 x 真伪的概率向量 $D(x) = \{p_{\text{realness}}(x, u); u \in \Omega\}$, 其中 Ω 是 p_{realness} 输出结果 u 的集合, 每一个 u 都可以被视为一种潜在的真实度量.

公式 (1) 中, 1 和 0 分别代表真实样本和生成样本的真实性度量; 而 RealnessGAN 则假设两个虚拟分布 A_1 和 A_0 是定义在 Ω 上的两个分布向量, 分别代表真实样本和生成样本的真实性向量. 其中, A_0 被设置为高斯分布, A_1 被设置为均匀分布. 利用 KL 散度和替换公式 (1) 中两个标量之间的差值, 则 G 和 D 的目标函数定义为:

$$\max_G \min_D V(G, D) = E_{x \sim p_{\text{data}}} [KL(A_1 \| D(x))] + E_{x \sim p_g} [KL(A_0 \| D(x))] \quad (3)$$

当 p_{realness} 的输出结果只有两种, 即 $\Omega = \{u_0, u_1\}$, 且令 $A_0(u_0) = A_1(u_1) = 1, A_0(u_1) = A_1(u_0) = 0$ 时, 公式 (3) 与公式 (2) 等价, 这表明公式 (2) 是公式 (3) 的一种特定情况. 并且文献 [26] 已经证明, 当固定 G 时, 对任一输出结果 u 和输入样本 x , 最优 D 满足 $D_G^*(x, u) = \frac{A_1(u)p_{\text{data}}(x) + A_0(u)p_g(x)}{p_{\text{data}}(x) + p_g(x)}$; 当 $D = D_G^*$, 存在一个输出结果 $u \in \Omega$, 使得 $A_1(u) \neq A_0(u)$ 时, G 最优当且仅当 $p_g = p_{\text{data}}$.

2 双向约束生成对抗网络

GAN 以及 RealnessGAN 都仅从一个方向拟合真实样本的数据分布, 其生成样本的数据分布往往与真实样本的数据分布存在较大偏差. 为了能够拟合更加真实的数据分布, 以提高生成图像的质量和多样性, 本文提出了一种双向约束生成对抗网络. 首先, 介绍了本文的网络架构设计; 然后, 对所提出的方法进行了推理论证, 通过数学推导证明了该方法的可行性; 最后, 详细阐述了 BCGAN 训练过程的相关算法.

2.1 网络架构设计

对于传统 GAN 变体而言, 理想情况下, 如图 1(b) 所示, 当且仅当真实样本与生成样本的数据分布相等, 即 $p_g = p_{\text{data}}$ 时, 目标函数 $V(G, D)$ 取最优值, 此时, 生成样本与真实样本真伪难辨. 现实问题中, p_{data} 是无法通过计算得到的. 因此, 研究者们希望通过优化 GAN 的网络架构和损失函数, 使得 p_g 接近于 p_{data} . GAN 变体通过博弈方法来拟合真实样本的数据分布, 直观地, 如图 1(c) 所示, p_g 和 p_{data} 重合部分越多, 表明生成样本的质量越高. 然而这些 GAN 变体在训练过程中, 仅考虑从一个方向去逼近真实样本的数据分布, 其生成样本的数据分布往往与真实样本的数据分布存在较大偏差, 以至于生成样本的质量和多样性普遍较低. 为此, 本文提出了一种双向约束生成对抗网络——BCGAN. 与传统 GAN 变体相比, BCGAN 在网络架构设计上增加了一个生成器模块, 通过训练使得两个生成样本的数据分布 p_{g1} 和 p_{g2} 从两个不同方向去逼近真实样本的数据分布 p_{data} , 如图 1(d) 所示.

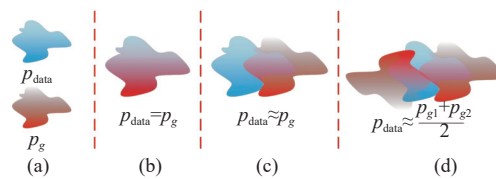


图 1 真实样本与生成样本的数据分布示例

由于在网络架构中增加了一个生成器模块, 因此, 稳定 BCGAN 模型训练过程需要的约束条件更苛刻. 为此, 本文通过一系列消融实验探讨了损失函数的约束项对模型训练过程的影响. 在 BCGAN 的训练过程中, 增加以下

优化策略: (1) 通过增加 $G_1(z)$ 和 $G_2(z)$ 数据分布之间的距离来丰富生成样本的多样性; (2) 通过减小 $D(G_1(z))$ 和 $D(G_2(z))$ 之间的差异来稳定训练过程, 提高生成样本的质量. BCGAN 与传统 GAN 变体的网络架构比较如图 2 所示, $\{\leftarrow\rightarrow\}$ 表示增加两者之间的距离, $\{\rightarrow\leftarrow\}$ 表示减小两者之间的距离.

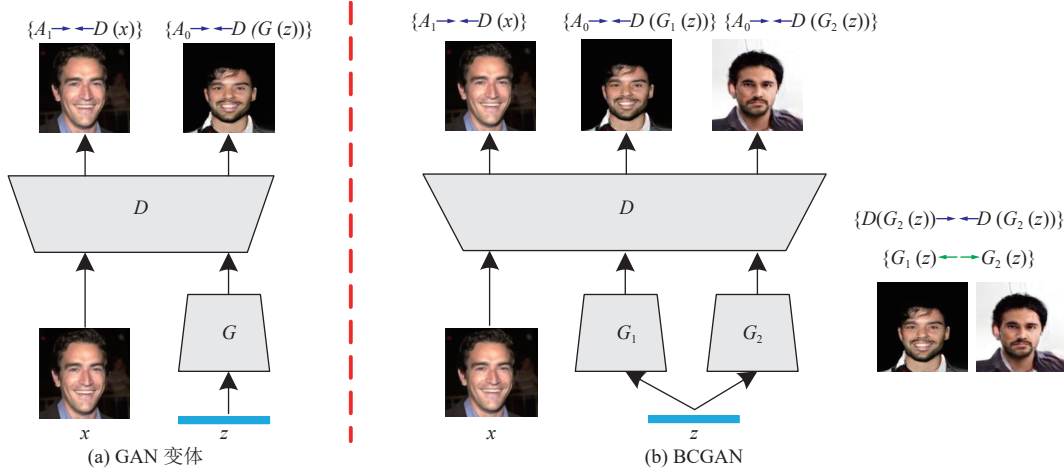


图 2 GAN 变体与 BCGAN 的网络架构比较

事实上, 本文方法能够应用于任何传统 GAN 变体模型, 本文选择以 RealnessGAN (2020) 模型^[26]为基础来设计 BCGAN 架构, 所有实验结果见第 3 节.

2.2 损失函数设计与证明

BCGAN 的训练过程可以视为一个极大极小值问题, 一方面, 鉴别器希望能够准确判别输入样本的真伪, 即输入真实样本判别为真, 而输入生成样本则判别为假; 另一方面, 两个生成器则希望从两个不同方向去逼近真实样本的数据分布, 以至于鉴别器无法准确判别输入样本的真伪. BCGAN 的目标函数定义如下:

$$\max_G \min_D V(G, D) = E_{x \sim p_{\text{data}}} [KL(A_1 \| D(x))] + E_{x \sim p_{g_1}} [KL(A_0 \| D(x))] + E_{x \sim p_{g_2}} [KL(A_0 \| D(x))] \tag{4}$$

其中, p_{data} 、 p_{g_1} 以及 p_{g_2} 分别为真实样本、生成样本 $G_1(z)$ 和 $G_2(z)$ 的数据分布.

首先希望从任意给定生成器 G 中寻找最优鉴别器 D 开始.

定理 1. 当固定 G 时, 对于任意输出结果 u 和输入样本 x , 最优鉴别器 D 满足:

$$D_G^*(x, u) = \frac{A_1(u) p_{\text{data}}(x) + A_0(u) p_{g_1}(x) + A_0(u) p_{g_2}(x)}{p_{\text{data}}(x) + p_{g_1}(x) + p_{g_2}(x)} \tag{5}$$

证明: 当固定 G 时, D 的目标函数为:

$$\begin{aligned} \min_D V(G, D) &= E_{x \sim p_{\text{data}}} [KL(A_1 \| D(x))] + E_{x \sim p_{g_1}} [KL(A_0 \| D(x))] + E_{x \sim p_{g_2}} [KL(A_0 \| D(x))] \\ &= \int_x p_{\text{data}}(x) \int_u A_1(u) \log \frac{A_1(u)}{D(x, u)} du dx \\ &\quad + \int_x p_{g_1}(x) \int_u A_0(u) \log \frac{A_0(u)}{D(x, u)} du dx + \int_x p_{g_2}(x) \int_u A_0(u) \log \frac{A_0(u)}{D(x, u)} du dx \\ &= - \int_x \int_u [p_{\text{data}}(x) A_1(u) + p_{g_1}(x) A_0(u) + p_{g_2}(x) A_0(u)] \log D(x, u) du dx \\ &\quad - \int_x [p_{\text{data}}(x) h(A_1) + p_{g_1}(x) h(A_0) + p_{g_2}(x) h(A_0)] dx \end{aligned} \tag{6}$$

其中, $h(A_1)$ 和 $h(A_0)$ 分别是 $A_1(u)$ 和 $A_0(u)$ 的熵. 公式 (6) 中第 2 项与 D 不相关, 记为 C_1 . 则公式 (6) 等价于:

$$\min_D V(G, D) = - \int_x C_2 \int_u p_x(u) \log D(x, u) du dx + C_1 \tag{7}$$

其中, $C_2 = p_{\text{data}}(x) + p_{g1}(x) + p_{g2}(x)$, $p_x(u) = \frac{p_{\text{data}}(x)A_1(u) + p_{g1}(x)A_0(u) + p_{g2}(x)A_0(u)}{p_{\text{data}}(x) + p_{g1}(x) + p_{g2}(x)}$. 然后有:

$$\begin{aligned} \min_D V(G, D) &= \int_x C_2 \left[- \int_u p_x(u) \log D(x, u) du - h(p_x) + h(p_x) \right] dx + C_1 \\ &= \int_x C_2 \left[- \int_u p_x(u) \log D(x, u) du + \int_u p_x(u) \log p_x(u) du \right] dx + \int_x C_2 h(p_x) dx + C_1 \\ &= \int_x C_2 KL(p_x(u) \| D(x, u)) dx + \int_x C_2 h(p_x) dx + C_1 \end{aligned} \quad (8)$$

其中, $h(p_x)$ 是 $p_x(u)$ 的熵, 公式 (8) 中后两项与 D 不相关. 因此, 当 $KL(p_x(u) \| D(x, u))$ 取最小值时, 鉴别器 D 达到最优, 即 $D_G^*(x, u) = p_x(u)$.

接下来讨论, 当 $D = D_G^*$ 时, 生成器 G 达到最优的条件.

定理 2. 当 $D = D_G^*$, 并且存在一个输出结果 $u \in \Omega$, 使得 $A_1(u) \neq A_0(u)$ 时, $V(G, D_G^*)$ 取最优值当且仅当 $p_{\text{data}}(x) = \frac{p_{g1}(x) + p_{g2}(x)}{2}$.

证明: 当 $p_{\text{data}}(x) = \frac{p_{g1}(x) + p_{g2}(x)}{2}$ 时, $D_G^*(x, u) = \frac{A_1(u) + 2A_0(u)}{3}$. 则:

$$V^*(G, D_G^*) = \int_x \int_u p_{\text{data}}(x) A_1(u) \log \frac{3A_1(u)}{A_1(u) + 2A_0(u)} dudx + \int_x \int_u [p_{g1}(x) + p_{g2}(x)] A_0(u) \log \frac{3A_0(u)}{A_1(u) + 2A_0(u)} dudx \quad (9)$$

记 $\Theta = p_{\text{data}}(x)A_1(u) + p_{g1}(x)A_0(u) + p_{g2}(x)A_0(u)$, 则对于 $V(G, D_G^*)$, 有:

$$\begin{aligned} V(G, D_G^*) &= \int_x \int_u p_{\text{data}}(x) A_1(u) \log \frac{A_1(u) [p_{\text{data}}(x) + p_{g1}(x) + p_{g2}(x)]}{A_1(u) p_{\text{data}}(x) + A_0(u) p_{g1}(x) + A_0(u) p_{g2}(x)} dudx \\ &\quad + \int_x \int_u p_{g1}(x) A_0(u) \log \frac{A_0(u) [p_{\text{data}}(x) + p_{g1}(x) + p_{g2}(x)]}{A_1(u) p_{\text{data}}(x) + A_0(u) p_{g1}(x) + A_0(u) p_{g2}(x)} dudx \\ &\quad + \int_x \int_u p_{g2}(x) A_0(u) \log \frac{A_0(u) [p_{\text{data}}(x) + p_{g1}(x) + p_{g2}(x)]}{A_1(u) p_{\text{data}}(x) + A_0(u) p_{g1}(x) + A_0(u) p_{g2}(x)} dudx \\ &= \int_x \int_u \{ p_{\text{data}}(x) A_1(u) \log A_1(u) + [p_{g1}(x) + p_{g2}(x)] A_0(u) \log A_0(u) \} dudx - \int_x \int_u \Theta \log p_x(u) dudx \end{aligned} \quad (10)$$

用 $V(G, D_G^*)$ 减去 $V^*(G, D_G^*)$ 得到:

$$\begin{aligned} V'(G, D_G^*) &= V(G, D_G^*) - V^*(G, D_G^*) = \int_x \int_u \Theta \log \frac{A_1(u) + 2A_0(u)}{3p_x(u)} dudx \\ &= - \int_x \int_u \Theta \log \frac{\Theta}{[A_1(u) + 2A_0(u)] [p_{\text{data}}(x) + p_{g1}(x) + p_{g2}(x)]} dudx \\ &= - KL \left(\Theta \parallel \frac{[A_1(u) + 2A_0(u)] [p_{\text{data}}(x) + p_{g1}(x) + p_{g2}(x)]}{3} \right) \end{aligned} \quad (11)$$

因为 $V^*(G, D_G^*)$ 是关于 G 的常数, 因此求 $V(G, D_G^*)$ 的最优值等价于求 $V'(G, D_G^*)$ 的最优值, 即:

$$\Theta = \frac{[A_1(u) + 2A_0(u)] [p_{\text{data}}(x) + p_{g1}(x) + p_{g2}(x)]}{3} \quad (12)$$

整理公式 (12), 得:

$$[A_1(u) - A_0(u)] \left[p_{\text{data}}(x) - \frac{p_{g1}(x) + p_{g2}(x)}{2} \right] = 0 \quad (13)$$

因此, 若存在一个输出结果 $u \in \Omega$ 使得 $A_1(u) \neq A_0(u)$, 则当且仅当 $p_{\text{data}}(x) = \frac{p_{g1}(x) + p_{g2}(x)}{2}$ 时 $V(G, D_G^*)$ 取最优值. 这一证明结果与图 1(d) 的思想相对应.

如理论分析所示, 鉴别器 D 的理想目标则是最小化 $KL(A_1 \| D(x))$ 和 $KL(A_0 \| D(G(z)))$. 而生成器 G 的理想目标

则是最大化 $KL(A_0\|D(G_1(z)))$ 和 $KL(A_0\|D(G_2(z)))$, 同时最小化 $KL(A_1\|D(G_1(z)))$ 和 $KL(A_1\|D(G_2(z)))$. 因此, 鉴别器 D 和生成器 G 的损失函数分别定义为:

$$loss_D = KL(A_1\|D(x)) + KL(A_0\|D(G_1(z))) + KL(A_0\|D(G_2(z))) \quad (14)$$

$$loss_G = -KL(A_0\|D(G_1(z))) + KL(A_1\|D(G_1(z))) - KL(A_0\|D(G_2(z))) + KL(A_1\|D(G_2(z))) \quad (15)$$

此外, 为了稳定训练过程, 希望减小 $D(G_1(z))$ 和 $D(G_2(z))$ 之间的差异. 因此, 加入约束项后, 鉴别器 D 的损失函数定义为:

$$loss_D = KL(A_1\|D(x)) + KL(A_0\|D(G_1(z))) + KL(A_0\|D(G_2(z))) + \lambda_D \|D(G_1(z)) - D(G_2(z))\| \quad (16)$$

其中, λ_D 为约束项系数.

同样地, 为了丰富生成样本的多样性, 还希望增加 $G_1(z)$ 和 $G_2(z)$ 数据分布之间的距离. 因此, 加入约束项后, 生成器 G 的损失函数定义为:

$$loss_G = -KL(A_0\|D(G_1(z))) + KL(A_1\|D(G_1(z))) - KL(A_0\|D(G_2(z))) + KL(A_1\|D(G_2(z))) - \lambda_G \|G_1(z) - G_2(z)\| \quad (17)$$

其中, λ_G 为约束项系数.

2.3 BCGAN 算法

本文方法训练采用 Adam^[27] 自适应算法, 其中, 一阶动量为 β_1 、二阶动量为 β_2 , 学习率为 l_r ; 鉴别损失函数约束项系数为 λ_D , 生成损失函数约束项系数为 λ_G . BCGAN 训练过程的相关算法描述如算法 1.

算法 1. BCGAN 算法.

输入: 批量数 $m=32$, 真实图像 x , 噪声向量 $z \sim N_{128}(0, I)$, 迭代次数 $Iterations = 520000$, 约束项系数 λ_D 和 λ_G , 鉴别器迭代次数 k_D , 生成器迭代次数 k_G , 虚拟分布 A_1 和 A_0 , 输出结果 Ω 的数量 n ;

输出: 生成图像 $G_1(z)$ 和 $G_2(z)$.

1) **for** $i = 1, 2, \dots, Iterations$ **do**

2) **for** $j = 1, \dots, k_D$ **do**

3) $gauss = N_{1000}(0, 0.1)$; $count, bins = histogram(gauss, n)$; $A_0 = \frac{count}{\sum count}$.

4) $unif = U_{1000}(-1, 1)$; $count, bins = histogram(unif, n)$; $A_1 = \frac{count}{\sum count}$.

5) Minibatch of m noise samples $\{z_1, z_2, \dots, z_m\}$ from noise prior $N_{128}(0, I)$.

6) Minibatch of m examples from $\{x_1, x_2, \dots, x_m\}$.

7) Update the discriminator:

$$loss_D = KL(A_1\|D(x)) + KL(A_0\|D(G_1(z))) + KL(A_0\|D(G_2(z))) + \lambda_D \|D(G_1(z)) - D(G_2(z))\|;$$

8) **end for**

9) **for** $k = 1, \dots, k_G$ **do**

10) Minibatch of m noise samples $\{z_1, z_2, \dots, z_m\}$ from noise prior $N_{128}(0, I)$.

11) Update the generator:

$$loss_G = -KL(A_0\|D(G_1(z))) + KL(A_1\|D(G_1(z))) - KL(A_0\|D(G_2(z))) + KL(A_1\|D(G_2(z))) - \lambda_G \|G_1(z) - G_2(z)\|.$$

12) **end for**

13) **end for**

3 实验结果分析

本节实验从多个方面对 BCGAN 的性能进行研究. 首先, 在一个合成数据集上研究了 BCGAN 对真实样本数

据分布的拟合能力. 然后, 在 3 个不同真实数据集上对 BCGAN 的生成样本进行了评估, 并且将 BCGAN 方法应用到了几种不同的骨干网络. 最后, 研究了损失函数中约束项对 BCGAN 性能的影响.

文献 [26] 中大量实验结果已经证明 RealnessGAN (2020) 所生成的图像质量优于其他传统 GAN 变体, 如 DCGAN (2015)^[6]、WGAN-GP (2017)^[21] 以及 LSGAN (2017)^[23] 等. 因此, 本文实验只以 RealnessGAN 作为基线, 与本文提出的 BCGAN 作对比.

3.1 实验环境及评估方法

本文所有实验均在 Ubuntu 18.04 系统上完成, 具体的实验环境如表 1 所示.

表 1 实验环境

实验环境	参数
PyCharm	2020.2
Python	3.6.12
PyTorch	1.4.0
CUDA	10.1
CPU	Inter Xeon(R) Silver 4210R CPU@2.40 GHz×20
GPU	NVIDIA TITAN RTX/PCIe/SSE2×1

在训练过程中, 即使不同的优化算法也会对 GAN 的性能产生显著影响, 同时通过对一些超参数的巧妙调整可以优化 GAN 的性能^[25]. 本文实验旨在验证从两个不同方向逼近真实样本的数据分布更有利于 GAN 模型拟合真实数据分布, 从而提升生成样本的质量和多样性. 因此, 为保证实验结果的公平性, 对于其他参数设置均与文献 [26] 中保持一致.

目前在 GAN 领域中, IS (inception score) 分数^[28] 和 FID (Fréchet inception distance) 分数^[29] 是两个最经典的性能评价指标, 几乎已成为该领域内通用的评价标准^[25]. 为了确保比较一致性, 本文所有实验结果均以 IS 分数作为定量评估生成样本多样性的指标, IS 分数是由 5 000 张生成样本计算得出; 以 FID 分数作为定量评估生成样本质量的指标, FID 分数是由 5 000 张生成样本和所有真实样本计算得出. 此外, 本文还比较了不同方法所占用的计算资源, 训练时间以及 SWD (sliced Wasserstein distance) 分数^[30]. SWD 分数以 Wasserstein 距离为标准, 是衡量两个数据分布之间差异的评估方法, SWD 分数越小表示两个数据分布之间的差异越小.

3.2 在合成数据集上的表现

在真实数据集上计算 p_{data} 通常是困难的, 为了可视化比较不同模型对 p_{data} 的拟合能力, 本文在一个合成数据集上进行了实验. 该合成数据集由 10 万个 2D 点组成, 这些 2D 点是从 9 个各向同性高斯分布的混合物中采样得到的, 其平均值排列在一个 3×3 的网格中, 方差为 0.05. 如图 3 红色虚线左侧所示, 该合成数据集上的数据分布 p_{data} 包含了 9 个分离良好的模态.

对于在合成数据集上的实验, 使用 Adam 优化器^[27] 对所有模型进行 5 万次迭代训练, 将学习率设置为 0.000 1, 超参数 β_1 设置为 0.8、 β_2 设置为 0.999, 鉴别器迭代次数 k_D 设置为 1, 生成器迭代次数 k_G 设置为 3, 输出结果 Ω 的数量 n 设置为 5. 此外, 将公式 (16) 中约束项系数 λ_D 和公式 (17) 中约束项系数 λ_G 都设置为 0, 即不考虑约束项的影响. 图 3 红色虚线右侧从第 1 行至第 3 行依次展示了使用 RealnessGAN、BCGAN 以及带有 3 个生成器的模型进行训练得到的生成样本数据分布. 相比于 RealnessGAN, BCGAN 恢复了 p_{data} 中间的模态形式, 且两个生成器通过训练能够分别学习到 p_{data} 不同的模态形式, 这表明 BCGAN 对 p_{data} 的拟合能力要优于 RealnessGAN. 由图 3 不难发现, BCGAN 拟合真实数据分布 p_{data} 的收敛速度更快. 此外, 从图 3 可以看出, 使用 3 个生成器进行训练得到了相似甚至低于 BCGAN 的结果. 利用 3 个生成器增加了额外的计算资源和训练时间, 因此在真实数据集上不再考虑使用 3 个或 3 个以上的生成器进行训练.

3.3 在真实数据集上的表现

本文在以下 3 个不同公开挑战数据集上对 BCGAN 的生成样本进行评估.

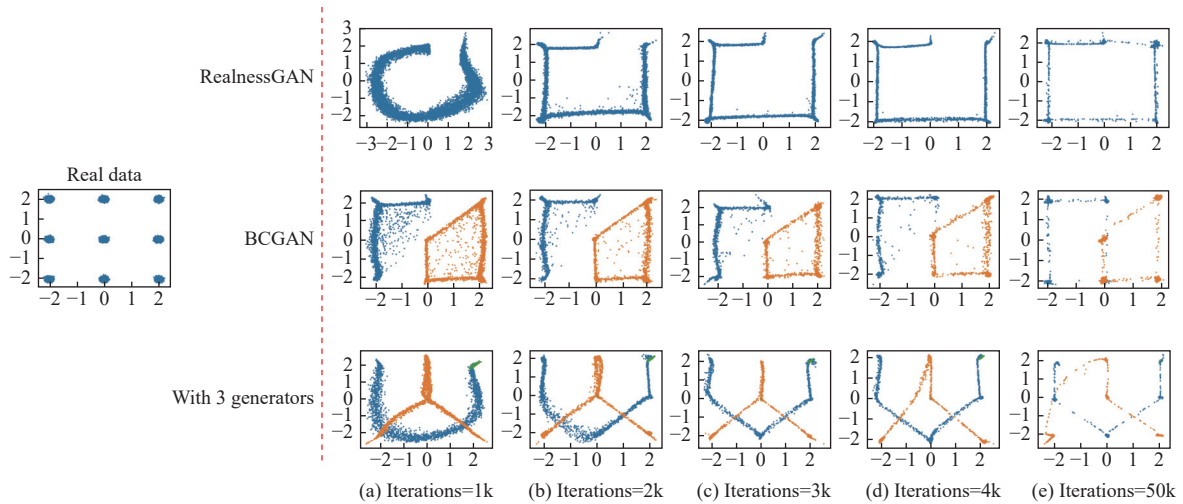


图3 合成数据集与生成样本的数据分布

CelebA 数据集^[31]共包含 202 599 张人脸图像. 将整个 CelebA 数据集为真实样本, 分别在 RealnessGAN 和 BCGAN 上进行训练.

Bedroom 数据集共包含 3 033 042 张卧室图像. Bedroom 数据集是 LSUN 数据集^[32]的一类子集, 从 Bedroom 数据集中随机选取 10% 的图像数据作为真实样本, 分别在 RealnessGAN 和 BCGAN 上进行训练.

Bridge 数据集共包含 818 687 张桥梁图像. Bridge 数据集也是 LSUN 数据集^[32]的一类子集, 从 Bridge 数据集中随机选取 50% 的图像数据作为真实样本, 分别在 RealnessGAN 和 BCGAN 上进行训练.

对于在真实数据集上的实验, 使用 Adam 优化器^[27]对所有模型进行 52 万次迭代训练, 将学习率设置为 0.000 2, 超参数 β_1 设置为 0.5、 β_2 设置为 0.9, 鉴别器迭代次数 k_D 设置为 1, 生成器迭代次数 k_G 设置为 1, 输出结果 Ω 的数量 n 设置为 51. 此外, 将公式 (16) 中约束项系数 λ_D 设置为 1, 约束项 $\|D(G_1(z)) - D(G_2(z))\|$ 的值由 $KL(D(G_1(z)) \| D(G_2(z)))$ 计算得到; 将公式 (17) 中约束项系数 λ_G 设置为 0.5, 约束项 $\|G_1(z) - G_2(z)\|$ 的值由 L_2 范数计算得到.

为了节省计算资源和训练时间, 所有生成样本的分辨率被限制在 256×256 . 图 4 展示了不同方法所生成的图像样本, 黑色虚线左侧的图像样本由 RealnessGAN 产生; 红色虚线左右两侧的图像样本分别由 BCGAN 的两个生成器产生. 从视觉效果上观察, BCGAN 所生成的图像样本更加真实. 并且, BCGAN 通过增加 $G_1(z)$ 和 $G_2(z)$ 数据分布之间的距离的确能够丰富生成样本的多样性, 例如在 CelebA 数据集上, 生成器 G_1 所生成的样本背景整体偏亮, 而生成器 G_2 所生成的样本背景整体偏暗.

对基线 RealnessGAN 和 BCGAN 在 3 个不同数据集上获得的 FID 和 IS 分数进行测量, 测量结果如表 2 所示. 其中, FID 和 IS 分数的最小值、最大值、平均值和标准差由训练过程中每迭代 2 万次, 3 万次, ..., 52 万次计算得到. 从表 2 不难看出, 在 CelebA、Bedroom 和 Bridge 这 3 个不同公开挑战数据集上, BCGAN 在 FID 分数这个定量评估指标上获得最好的分数分别为 21.83、29.42 和 42.10, 较基线 RealnessGAN 相比, 分别提升了 0.76、5.94 和 5.50; 在 3 个数据集上, 较基线 RealnessGAN 相比, BCGAN 在 IS 分数这个定量评估指标上也获得了更好的评分. 实验证明, 本文提出的 BCGAN 方法能够有效提升生成样本的质量和多样性. 此外, BCGAN 在 FID 和 IS 分数这两个定量评估指标上的标准差均不高于 RealnessGAN, 这表明 BCGAN 的训练过程更加平滑稳定.

基线 RealnessGAN 和 BCGAN 在 CelebA 数据集上得到的训练曲线如图 5 所示, 其中图 5(a) 和图 5(b) 分别表示 FID 训练曲线和 IS 训练曲线. 由图 5(a) 和图 5(b) 可以看出, 在最后 10 万次迭代过程中, BCGAN 的 FID 分数和 IS 分数总体上都要优于 RealnessGAN, 这表明利用 BCGAN 在 CelebA 数据集上训练得到的生成样本在图像质量和多样性方面都要优于基线 RealnessGAN.

基线 RealnessGAN 和 BCGAN 在 Bedroom 数据集上得到的训练曲线如图 6 所示. 图 6(a) 表示 FID 训练曲线, 较 RealnessGAN 相比, BCGAN 的 FID 训练曲线要更加稳定, 并且利用 BCGAN 方法得到了更低的 FID 分数; 图 6(b)

表示 IS 训练曲线, 从图 6(b) 可以看出, RealnessGAN 和 BCGAN 均在第 4 万次迭代过程中得到了最好的 IS 分数, 而在最后 10 万次迭代过程中, BCGAN 的 IS 分数总体上都要高于 RealnessGAN. 从对图 6(a) 和图 6(b) 的实验结果分析可以看出, BCGAN 方法在 Bedroom 数据集上表现更佳.

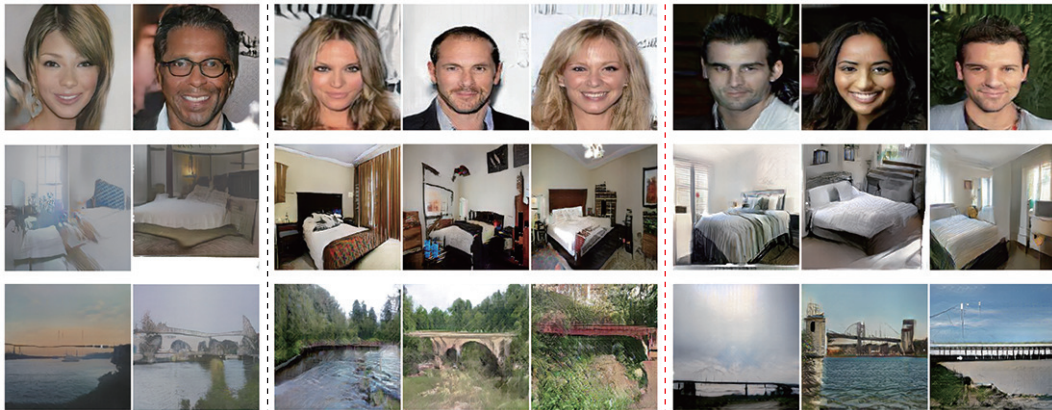


图 4 不同方法所生成的图像样本

表 2 FID 和 IS 的最小值、最大值、平均值以及标准差

数据集	方法	FID↓				IS↑			
		最小值	最大值	平均值	标准差	最小值	最大值	平均值	标准差
CelebA	RealnessGAN	22.59	92.55	30.34	13.26	2.38	2.76	2.61	0.07
	BCGAN	21.83	64.69	29.44	9.19	2.40	2.80	2.67	0.07
Bedroom	RealnessGAN	35.36	174.25	52.54	26.23	2.82	3.82	3.37	0.17
	BCGAN	29.42	144.61	44.01	21.75	3.22	3.82	3.40	0.16
Bridge	RealnessGAN	47.60	171.30	66.99	24.68	2.94	4.98	4.60	0.42
	BCGAN	42.10	141.17	60.57	21.97	2.75	5.04	4.68	0.40

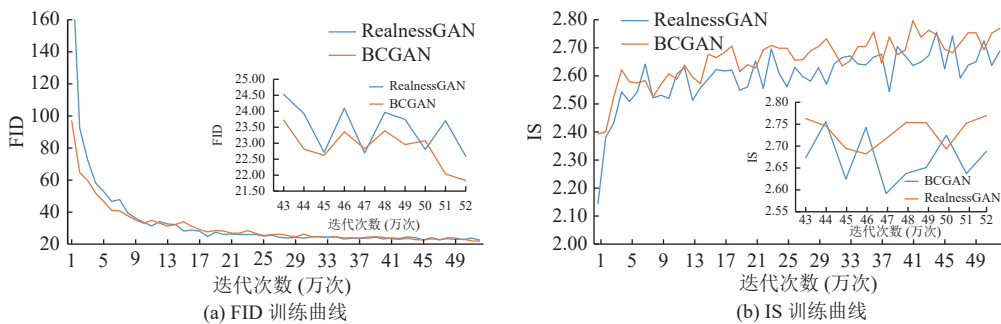


图 5 利用不同方法在 CelebA 数据集上得到的 FID 和 IS 训练曲线

基线 RealnessGAN 和 BCGAN 在 Bridge 数据集上得到的训练曲线如图 7 所示. 较 RealnessGAN 相比, BCGAN 的 FID 训练曲线和 IS 训练曲线都更加平稳, 并且在最后 10 万次迭代过程中, BCGAN 的 FID 分数和 IS 分数总体上都要优于 RealnessGAN. 实验证明, 较基线 RealnessGAN 相比, 利用 BCGAN 方法能够得到图像质量更高且多样性更丰富的生成样本.

由于 BCGAN 增加了一个生成器模块, 其所需要的计算资源以及训练时间要大于 RealnessGAN. 不过 RealnessGAN 是基于 DCGAN 的架构设计的, 因此, 增加一个生成器模块所占用的计算资源是比较小的. 本文对不同方法所占用的计算资源 (GPU 内存), 训练时间 (一次迭代时间) 以及得到的 SWD 分数进行了比较, 实验结果如表 3 所示, 其中 SWD 分数由第 52 万次迭代训练的生成图像计算得到. 由于 3 个数据集集中的图像分辨率均被裁剪

为 256×256 , 且每次迭代所使用的最小批量数均为 32 张, 因此同一种方法在不同数据集下所占用的计算资源和一次迭代时间是相等的. 从表 3 可以看出, 尽管 BCGAN 所需要的计算资源以及训练时间稍微高于 RealnessGAN, 但是 BCGAN 取得了远低于 RealnessGAN 的 SWD 分数, 即 BCGAN 所生成图像的数据分布更加接近于真实样本的数据分布.

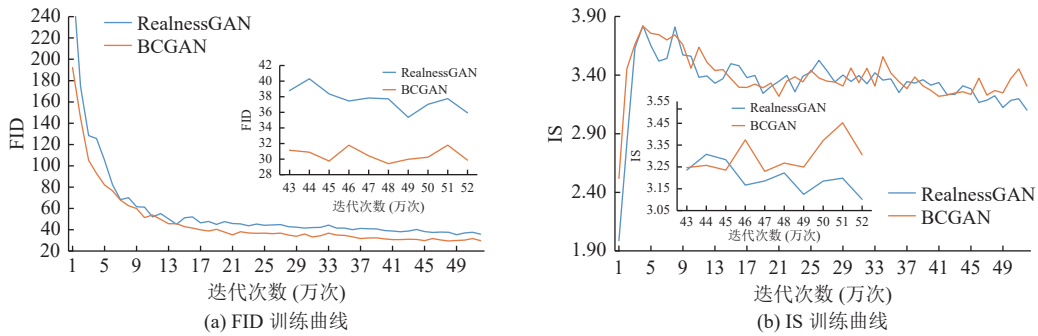


图 6 利用不同方法在 Bedroom 数据集上得到的 FID 和 IS 训练曲线

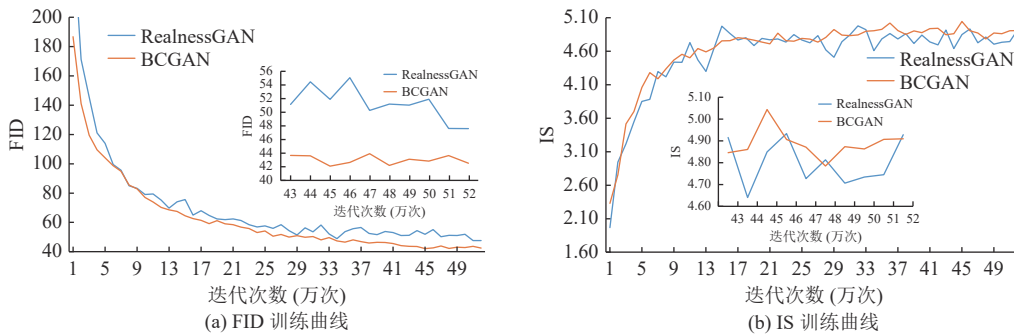


图 7 利用不同方法在 Bridge 数据集上得到的 FID 和 IS 训练曲线

表 3 不同方法所占用计算资源, 训练时间以及 SWD 分数的比较结果

方法	CelebA		$10^3 \times \text{SWD}$	Bedroom	Bridge
	GPU内存 (GB)↓	一次迭代时间 (s)↓		$10^3 \times \text{SWD}$	$10^3 \times \text{SWD}$
RealnessGAN	2.53	0.17	15.80	37.02	26.02
BCGAN	3.00	0.24	8.96	10.60	9.35

此外, 本文在 CelebA 数据集上, 增加了使用两种方法进行训练的迭代次数, 并且计算了它们的 FID 曲线, 实验结果如图 8 所示. 其中, 垂直的绿色虚线表示第 52 万次迭代结果, 此时 RealnessGAN 和 BCGAN 得到的 FID 分数分别为 22.59 和 21.83. 对两种方法继续进行训练迭代至 100 万次, RealnessGAN 和 BCGAN 所得到的最低 FID 分数分别为 21.91 和 20.21. 这表明即使迭代 100 万次, RealnessGAN 所得到的最好的 FID 分数仍然小于 21.83, 证明了本文方法的优越性. 并且随着迭代次数的增加, RealnessGAN 的 FID 分数渐渐趋于稳定, 而 BCGAN 的 FID 分数仍然具有下降的空间.

本文从视觉效果上, 对两种方法在 CelebA 数据集上的生成过程进行了比较, 实验结果如图 9 所示, 黑色虚线上侧表示由 RealnessGAN 所生成的图像样本; 红色虚线上下两侧的图像样本分别由 BCGAN 的两个生

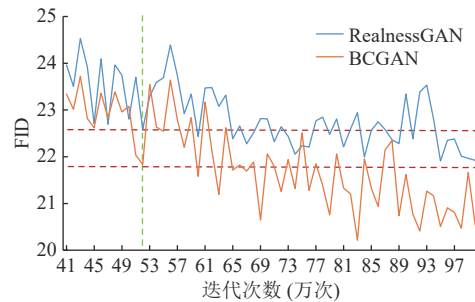


图 8 比较不同方法的 FID 训练曲线

成器产生. 如图 9 所示, BCGAN 的生成过程相对更加稳定, 例如在生成过程中, RealnessGAN 生成样本的嘴巴时而张开时而闭合, 而 BCGAN 两个生成样本的嘴巴状态基本保持一致.

最后, 本文在 CelebA 数据集上对所提出的方法在几种骨干网络上的表现进行了实验验证, 实验结果如表 4 所示, 其中 IS、FID 以及 SWD 分数由第 52 万次迭代训练的生成图像计算得到. 第 1 组实验以 DCGAN 的生成器和鉴别器作为骨干网络进行训练; 第 2 组实验将 DCGAN 生成器替换为一个基于残差网络的生成器^[33], 该生成器是在 DCGAN 生成器上加入两个残差块得到的; 第 3 组实验将鉴别器替换为 PatchGAN 鉴别器^[34]. 从表 4 中可以看出, 对于不同的骨干网络, 本文方法在 3 种定量评价指标上所得到的结果都要优于 RealnessGAN. 不过, 本文方法也为此消耗了更多的计算资源和训练时间. 值得注意的是, 相对于第 1 组实验, 第 2 组实验需要更多的 GPU 内存和训练时间, 然而得到的实验结果却不如第 1 组实验, 这可能是由于生成器和鉴别器的性能不匹配所导致的, 一个强大的生成器往往会导致在训练过程中鉴别器容易发生拟合. 因此, 构建一个性能对等的生成器和鉴别器是重要的.

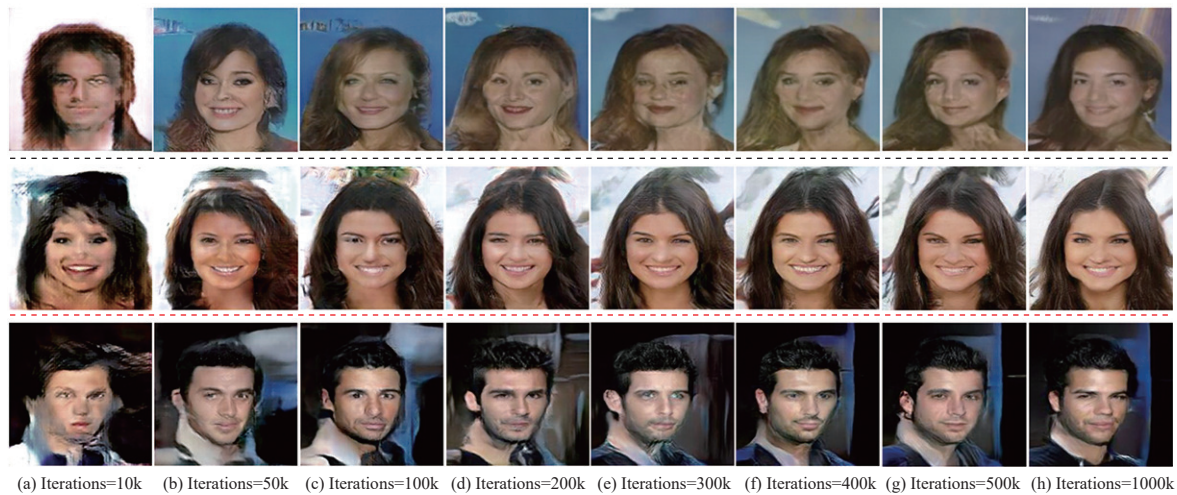


图 9 不同方法的生成过程展示

表 4 本文方法在几种骨干网络上的应用

骨干网络	方法	IS↑	FID↓	$10^3 \times \text{SWD} \downarrow$	GPU内存 (GB)↓	一次迭代时间 (s)↓
DCGAN生成器和DCGAN鉴别器	RealnessGAN	2.69	22.59	15.80	2.53	0.17
	BCGAN	2.77	21.83	8.96	3.00	0.24
基于残差网络的生成器 ^[33] 和DCGAN鉴别器	RealnessGAN	2.76	25.29	20.13	2.95	0.24
	BCGAN	2.82	22.11	11.81	3.84	0.33
基于残差网络的生成器 ^[33] 和PatchGAN鉴别器 ^[34]	RealnessGAN	2.88	21.52	13.33	4.51	0.37
	BCGAN	3.02	19.27	7.98	5.40	0.47

3.4 消融实验

为简化实验过程, 本节选择在 CelebA 数据集上进行消融实验, 以此评估 BCGAN 不同约束项的重要性. 对于鉴别器的约束项 $\|D(G_1(z)) - D(G_2(z))\|$, 它是由 $KL(D(G_1(z)) \| D(G_2(z)))$ 计算得到的, 因此将其约束项系数 λ_D 设置为 1, 以保证和鉴别器其他损失项系数相一致; 而对于生成器的约束项 $\|G_1(z) - G_2(z)\|$, 它的值由 L_2 范数计算得到, 因此将其约束项系数 λ_G 设置为 0.5, 以保证和生成器其他损失项的约束力度相匹配. 本文考虑了几种极端情况, 依次对两个损失函数不进行约束, 表 5 展示了修改约束项系数对 FID 和 IS 分数的影响. 其中, FID 和 IS 分数为训练过程中每迭代 2 万次, 3 万次, ..., 52 万次计算得到的最优值. BCGAN 中鉴别器 D 和生成器 G 的损失函数分别如

公式 (16) 和公式 (17) 所示, 当两个损失函数均不加约束项, 即约束项系数 $\lambda_D = \lambda_G = 0$ 时, 得到的 FID 分数和 IS 分数最差, 分别为 85.93 和 2.62; 当仅约束生成器 G 的损失函数, 即约束项系数 $\lambda_D = 0$ 且 $\lambda_G = 0.5$ 时, 得到了最好的 IS 分数, 这表明通过增加 $G_1(z)$ 和 $G_2(z)$ 数据分布之间的距离能够丰富生成样本的多样性; 当仅约束鉴别器 D 的损失函数, 即约束项系数 $\lambda_D = 1$ 且 $\lambda_G = 0$ 时, 得到的 FID 分数为 23.30, 远远低于未约束鉴别器 D 损失函数时得到的 FID 分数, 这表明通过减小 $D(G_1(z))$ 和 $D(G_2(z))$ 之间的差异能够稳定训练过程, 提高生成样本的质量; 而当对鉴别器 D 和生成器 G 的损失函数都进行约束, 即约束项系数 $\lambda_D = 1$ 且 $\lambda_G = 0.5$ 时, 得到了最好的 FID 分数, 且此时的 IS 分数为 2.80, 均优于基线 RealnessGAN 在 CelebA 数据集上的表现。

表 5 约束项系数的修改对 FID 和 IS 分数的影响

约束项系数	无约束	仅约束生成器	仅约束鉴别器	双向约束
λ_D	0	0	1	1
λ_G	0	0.5	0	0.5
FID↓	85.93	44.18	23.30	21.83
IS↑	2.62	2.84	2.79	2.80

不同约束项系数对应的 FID 和 IS 训练曲线如图 10 所示. 当两个损失函数均不加约束项, 即约束项系数 $\lambda_D = \lambda_G = 0$ 时, BCGAN 在 CelebA 数据集上的训练过程会出现模型崩溃的问题; 而当对鉴别器 D 和生成器 G 的损失函数都进行约束, 即约束项系数 $\lambda_D = 1$ 且 $\lambda_G = 0.5$ 时, 得到了收敛更快且更加平稳的训练曲线, 这表明鉴别器 D 和生成器 G 损失函数的两个约束项能够稳定 BCGAN 的训练过程。

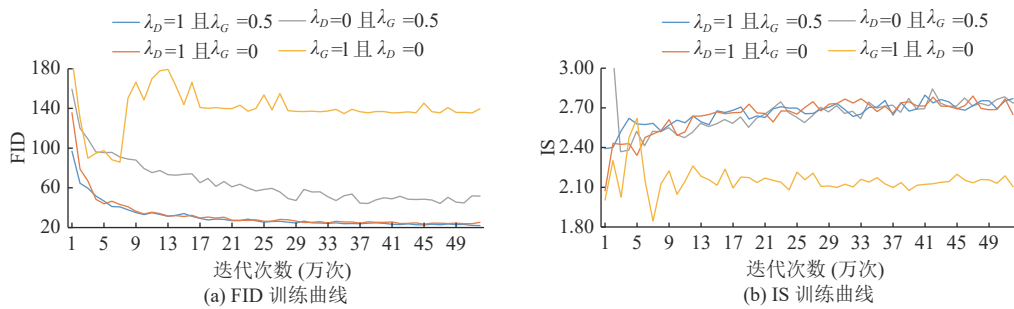


图 10 不同约束项系数对应的 FID 和 IS 训练曲线

本文还在合成数据集上使用不同约束项系数得到的生成结果进行了实验, 实验结果如图 11 所示. 其中, 红色虚线右侧第 1 行为两个损失函数均不加约束项, 即约束项系数 $\lambda_D = \lambda_G = 0$ 时所得到的实验结果, 这表明对于简单的合成数据集, 即使不使用约束项, 也能够很好地拟合真实样本的数据分布; 红色虚线右侧第 2 行为仅约束生成器 G 的损失函数, 即约束项系数 $\lambda_D = 0$ 且 $\lambda_G = 0.001$ 时所得到的实验结果, 此时两个生成器所学习到的数据分布差异最明显; 红色虚线右侧后两行分别表示仅约束鉴别器 D 的损失函数以及对鉴别器 D 和生成器 G 的损失函数都进行约束所得到的实验结果, 这两种约束方法在合成数据集上得到了相似的实验结果。

最后, 本文对几种不同变体在合成数据集上的表现进行了比较. 实验结果如图 12 所示, 其中图 12(a) 表示真实数据分布; 图 12(b) 是在 RealnessGAN 上得到的数据分布; 图 12(c) 是在一次迭代中输入两个不同向量得到的数据分布; 图 12(d) 是使用 3 个生成器模块得到的数据分布; 图 12(e) 是本文方法得到的数据分布. 在一个生成器中输入两个不同向量 z 和 $z + \Delta z$, 来控制两个不同的输出结果 $G(z)$ 和 $G(z + \Delta z)$, 得到了与 RealnessGAN 类似的生成样本数据分布. 而使用 3 个生成器进行训练所得到的生成样本数据分布虽然也恢复了 p_{data} 中间的模式形式, 但是用 3 个生成器会消耗更多的计算资源和训练时间. 因此, 本文所提出的 BCGAN 方法对真实数据分布的拟合能力更具优势。

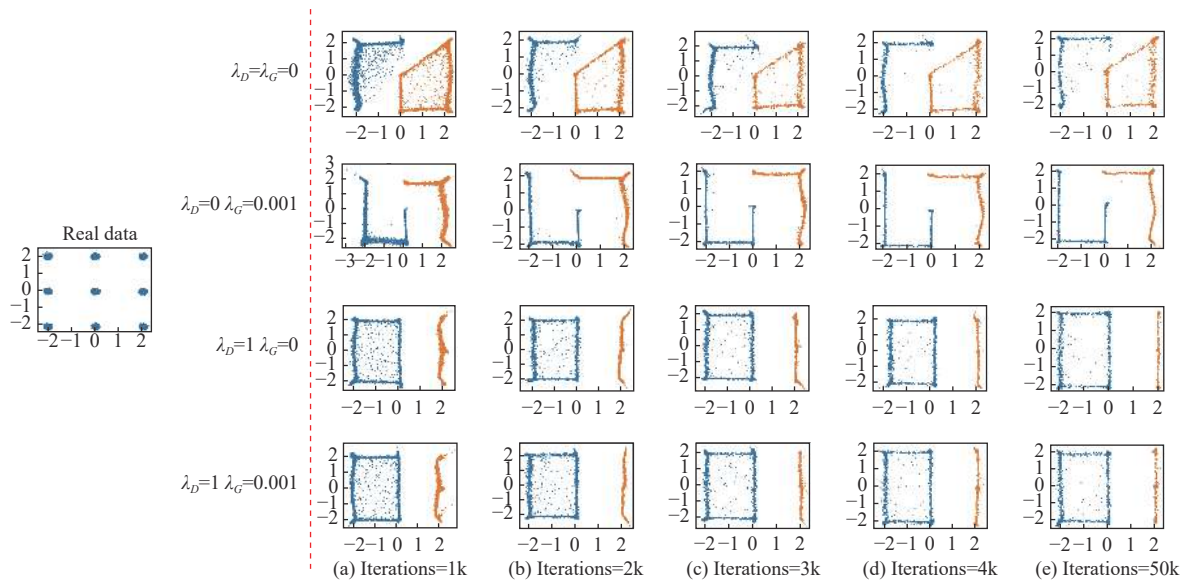


图 11 不同约束项系数在合成数据集上的表现

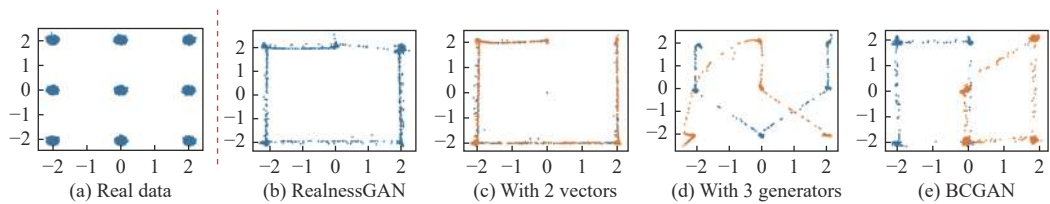


图 12 比较不同方法的数据分布

4 结束语

本文首次考虑从两个不同方向去逼近真实样本的数据分布, 在训练过程中, 一方面通过增加两个生成样本数据分布之间的距离来丰富生成样本的多样性, 另一方面通过减小鉴别器对两个生成样本数据分布之间的差异来稳定训练过程, 提高生成样本的质量. 本文在一个合成数据集上研究了 BCGAN 对真实样本数据分布的拟合能力, 实验证明 BCGAN 的拟合能力要优于基线 RealnessGAN, 且收敛速度更快. 然后在 CelebA、Bedroom 和 Bridge 这 3 个不同公开挑战数据集上进行了实验, 一系列实验证明, 较基线 RealnessGAN 相比, 本文提出的 BCGAN 方法能够有效提升生成样本的质量和多样性. 此外, BCGAN 的训练过程更加平稳.

本文通过实验证明了 BCGAN 能够有效提升生成样本的质量和多样性. 之后的研究工作一方面尝试利用一个冻结的预训练生成模型 F 和一个浅层的可训练网络 C 来设计生成模块 $G (G = C \circ F)$, 希望进一步降低计算资源和训练时间; 另一方面尝试将该思想引入到其他生成任务, 如条件生成以及图像转换等, 以期得到更好的实验结果.

References:

- [1] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Proc. of the 27th Int'l Conf. on Neural Information Processing Systems. Montreal: ACM, 2014. 2672–2680. [doi: 10.5555/2969033.2969125]
- [2] Hu MF, Zuo X, Liu JW. Survey on deep generative model. Acta Automatica Sinica, 2022, 48(1): 40–74 (in Chinese with English abstract). [doi: 10.16383/j.aas.c190866]
- [3] Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: ICLR, 2017.
- [4] Miyato T, Kataoka T, Koyama M, Yoshida Y. Spectral normalization for generative adversarial networks. In: Proc. of the 6th Int'l Conf.

- on Learning Representations. Vancouver: ICLR, 2018.
- [5] Chen FJ, Zhu F, Wu QX, Hao YM, Wang ED, Cui YG. A survey about image generation with generative adversarial nets. *Chinese Journal of Computers*, 2021, 44(2): 347–369 (in Chinese with English abstract). [doi: [10.11897/SP.J.1016.2021.00347](https://doi.org/10.11897/SP.J.1016.2021.00347)]
- [6] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. In: Proc. of the 4th Int'l Conf. on Learning Representations. San Juan: ICLR, 2016.
- [7] Mirza M, Osindero S. Conditional generative adversarial nets. arXiv:1411.1784, 2014.
- [8] Chen X, Duan Y, Houthoofd R, Schulman J, Sutskever I, Abbeel P. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems. Barcelona: ACM, 2016. 2180–2188. [doi: [10.5555/3157096.3157340](https://doi.org/10.5555/3157096.3157340)]
- [9] Li X, Yan Y, Liu MH, Liu M. A generation adversarial network based on multi-condition confrontation and gradient optimization. *Journal of University of Electronic Science and Technology of China*, 2021, 50(5): 754–760 (in Chinese with English abstract). [doi: [10.12178/1001-0548.2020415](https://doi.org/10.12178/1001-0548.2020415)]
- [10] Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B. Adversarial autoencoders. arXiv:1511.05644, 2015.
- [11] Donahue J, Krähenbühl P, Darrell T. Adversarial feature learning. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: ICLR, 2017.
- [12] Ulyanov D, Vedaldi A, Lempitsky VS. It takes (only) two: Adversarial generator-encoder networks. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence, the 30th Innovative Applications of Artificial Intelligence, and the 8th AAAI Symp. on Educational Advances in Artificial Intelligence. New Orleans: AAAI, 2018. 1250–1257.
- [13] Larsen ABL, Sønderby SK, Larochelle H, Winther O. Autoencoding beyond pixels using a learned similarity metric. In: Proc. of the 33rd Int'l Conf. on Machine Learning. New York: ACM, 2016. 1558–1566. [doi: [10.5555/3045390.3045555](https://doi.org/10.5555/3045390.3045555)]
- [14] Jia XY, Zhang WZ, Li WW, Huang ZQ. Feature representation method for heterogeneous defect prediction based on variational autoencoders. *Ruan Jian Xue Bao/Journal of Software*, 2021, 32(7): 2204–2218 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6257.htm> [doi: [10.13328/j.cnki.jos.006257](https://doi.org/10.13328/j.cnki.jos.006257)]
- [15] Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: ICLR, 2019.
- [16] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 4396–4405. [doi: [10.1109/CVPR.2019.00453](https://doi.org/10.1109/CVPR.2019.00453)]
- [17] Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T. Analyzing and improving the image quality of StyleGAN. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 8107–8116. [doi: [10.1109/CVPR42600.2020.00813](https://doi.org/10.1109/CVPR42600.2020.00813)]
- [18] Skorokhodov I, Sotnikov G, Elhoseiny M. Aligning latent and image spaces to connect the unconnectable. In: Proc. of the 2021 IEEE/CVF Int'l Conf. on Computer Vision. Montreal: IEEE, 2021. 14124–14133. [doi: [10.1109/ICCV48922.2021.01388](https://doi.org/10.1109/ICCV48922.2021.01388)]
- [19] Nowozin S, Cseke B, Tomioka R. f -GAN: Training generative neural samplers using variational divergence minimization. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems. Barcelona: ACM, 2016. 271–279. [doi: [10.5555/3157096.3157127](https://doi.org/10.5555/3157096.3157127)]
- [20] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: ACM, 2017. 214–223. [doi: [10.5555/3305381.3305404](https://doi.org/10.5555/3305381.3305404)]
- [21] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved training of Wasserstein GANs. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: ACM, 2017. 5769–5779. [doi: [10.5555/3295222.3295327](https://doi.org/10.5555/3295222.3295327)]
- [22] Zhao JJ, Mathieu M, LeCun Y. Energy-based generative adversarial networks. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: ICLR, 2017.
- [23] Mao XD, Li Q, Xie HR, Lau RYK, Wang Z, Smolley SP. Least squares generative adversarial networks. In: Proc. of the 2017 IEEE Int'l Conf. on Computer Vision. Venice: IEEE, 2017. 2813–2821. [doi: [10.1109/ICCV.2017.304](https://doi.org/10.1109/ICCV.2017.304)]
- [24] Mao XD, Li Q, Xie HR, Lau RYK, Wang Z, Smolley SP. On the effectiveness of least squares generative adversarial networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2019, 41(12): 2947–2960. [doi: [10.1109/TPAMI.2018.2872043](https://doi.org/10.1109/TPAMI.2018.2872043)]
- [25] Tan HW, Wang GD, Zhou LY, Zhang ZL. Generative adversarial networks based on penalty of conditional entropy distance. *Ruan Jian Xue Bao/Journal of Software*, 2021, 32(4): 1116–1128 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6156.htm> [doi: [10.13328/j.cnki.jos.006156](https://doi.org/10.13328/j.cnki.jos.006156)]
- [26] Xiangli YB, Deng YB, Dai B, Loy CC, Lin DH. Real or not real, that is the question. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: ICLR, 2020.
- [27] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Proc. of the 3rd Int'l Conf. on Learning Representations. San Diego:

- ICLR, 2015.
- [28] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X. Improved techniques for training GANs. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems. Barcelona: ACM, 2016. 2234–2242. [doi: [10.5555/3157096.3157346](https://doi.org/10.5555/3157096.3157346)]
- [29] 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: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: ACM, 2017. 6629–6640. [doi: [10.5555/3295222.3295408](https://doi.org/10.5555/3295222.3295408)]
- [30] Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of GANs for improved quality, stability, and variation. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: ICLR, 2018.
- [31] Liu ZW, Luo P, Wang XG, Tang XO. Deep learning face attributes in the wild. In: Proc. of the 2015 IEEE Int'l Conf. on Computer Vision. Santiago: IEEE, 2015. 3730–3738. [doi: [10.1109/ICCV.2015.425](https://doi.org/10.1109/ICCV.2015.425)]
- [32] Yu F, Seff A, Zhang YD, Song SR, Funkhouser T, Xiao JX. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv:1506.03365, 2015.
- [33] Johnson J, Alahi A, Li FF. Perceptual losses for real-time style transfer and super-resolution. In: Proc. of the 14th European Conf. on Computer Vision. Amsterdam: Springer, 2016. 694–711. [doi: [10.1007/978-3-319-46475-6_43](https://doi.org/10.1007/978-3-319-46475-6_43)]
- [34] Isola P, Zhu JY, Zhou TH, Efros AA. Image-to-image translation with conditional adversarial networks. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 5967–5976. [doi: [10.1109/CVPR.2017.632](https://doi.org/10.1109/CVPR.2017.632)]

附中文参考文献:

- [2] 胡铭菲, 左信, 刘建伟. 深度生成模型综述. 自动化学报, 2022, 48(1): 40–74. [doi: [10.16383/j.aas.c190866](https://doi.org/10.16383/j.aas.c190866)]
- [5] 陈佛计, 朱枫, 吴清潇, 郝颖明, 王恩德, 崔芸阁. 生成对抗网络及其在图像生成中的应用研究综述. 计算机学报, 2021, 44(2): 347–369. [doi: [10.11897/SP.J.1016.2021.00347](https://doi.org/10.11897/SP.J.1016.2021.00347)]
- [9] 李响, 严毅, 刘明辉, 刘明. 基于多条件对抗和梯度优化的生成对抗网络. 电子科技大学学报, 2021, 50(5): 754–760. [doi: [10.12178/1001-0548.2020415](https://doi.org/10.12178/1001-0548.2020415)]
- [14] 贾修一, 张文舟, 李伟滢, 黄志球. 基于变分自编码器的异构缺陷预测特征表示方法. 软件学报, 2021, 32(7): 2204–2218. <http://www.jos.org.cn/1000-9825/6257.htm> [doi: [10.13328/j.cnki.jos.006257](https://doi.org/10.13328/j.cnki.jos.006257)]
- [25] 谭宏卫, 王国栋, 周林勇, 张自力. 基于一种条件熵距离惩罚的生成式对抗网络. 软件学报, 2021, 32(4): 1116–1128. <http://www.jos.org.cn/1000-9825/6156.htm> [doi: [10.13328/j.cnki.jos.006156](https://doi.org/10.13328/j.cnki.jos.006156)]



苟瑶(1997—), 男, 博士生, 主要研究领域为图像生成, 目标识别.



何玉杰(1987—), 男, 博士, 讲师, 主要研究领域为计算机视觉, 机器学习.



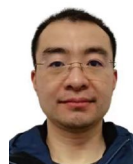
李敏(1971—), 女, 博士, 教授, 博士生导师, 主要研究领域为人工智能, 信息安全.



吴肇青(1997—), 男, 硕士, 主要研究领域为图像生成, 目标检测.



杜卫东(1988—), 男, 博士生, 主要研究领域为图像生成, 隐私保护.



宋雨(1982—), 男, 博士生, 主要研究领域为图像生成, 目标检测.