Logo Synthesis and Manipulation with Clustered Generative Adversarial Networks

Alexander Sage, Eirikur Agustsson, Radu Timofte, and Luc Van Gool

摘要

为一个新品牌设计一个标志是设计师和客户之间一个漫长而乏味的来回过程。在本文中，我们探讨了机器学习在多大程度上可以解决设计师的创造性任务。为此，我们建立了一个数据集- LLD -从万维网抓取的 600k+ 徽标。在这种多模态数据上训练生成对抗网络 (GANs) 进行标志合成并不简单，并且对一些最先进的方法会导致模式崩溃。我们建议使用通过聚类获得的合成标签来解缠和稳定 GAN 训练，并在 CIFAR10 和 ImageNet-small 上验证该方法以证明其通用性。我们能够产生一个高度多样的合理的标志，并展示潜在的空间探索技术，以一种互动的方式缓解标志设计任务。GANs 可以通过聚类获得合成标签来处理多模态数据，我们的研究结果显示了这种技术在标志合成和操作方面的创新潜力。

关键词： Logo Synthesis；GANs；Cluster；Latent Space Exploration

1 引言

对于客户和设计师来说，为一个新品牌设计一个标志通常是一个漫长而乏味的过程。许多最终未使用的草稿被制作出来，客户从中选择他最喜欢的，然后经过多个周期的改进，以满足客户的需求和愿望。特别是对于那些对最终产品没有具体想法的客户来说，这不仅会耗费时间，而且还会耗费大量成本。该课题提供了一个可以合成各种 Logo（生成的一些 Logo 例子如图 1所示）的深度学习框架，以此来加快设计师对客户所提需求的 Logo 设计任务。因此，客户可以根据形状和颜色等特定参数修改原型 Logo，或者将其向另一个原型 Logo 的特征进行一定程度的转移。根据客户所提特征，该系统可以帮助设计师和客户获得一个潜在的标志的想法，设计师可以在此基础上进行设计，省去了设计一个 Logo 需要打大量草稿的时间。

图 1. LLD 的原始 Logo（前三行）和 iWGAN-LC 生成的 Logo（后三行）

2 相关工作

在该课题的相关工作中，作者们在他们自己的多模态 Logo 数据中训练 GANs，作为用户自己实现人工 Logo 合成的第一步，该课题的主要贡献及其相关工作是：（1）构建了 LLD - 60 万 + 的大型 Logo 图像数据集；（2）提出了一种在多模态数据上稳定训练 GANs 模型的方法；（3）Logo 合成中 GANs 潜在空间的探索；（4）演示该课题提出的 Logo 合成方法结合界面应用程序辅助标志设计。

2.1 构建 Logo 图像数据集

现有的研究文献主要工作集中在检索、检测和识别较少数量的 Logo [6,8,15,18,20,27]，这些工作引入了许多数据集，最具代表性的大型公共 Logo 数据集如表 1前五行所示。但是，这些数据集不适合用于学习和验证自动标识发生器。同时，一些网页允许 (付费) 访问大量的图标，如 iconsdb.com(4135+ 图标)，icons8.com (59900+)，iconfinder.com (7473+)，iconarchive.com (450k+) 和 thenounproject.com (1m+)。由于大型公共 Logo 数据集中所包含的 Logo 的多样性较低，该课题从互联网上爬取了一个高度多样化的数据集-大型徽标数据集 (LLD)，如表 1后三行所示，该课题的 LLD 比迄今为止最大的公共徽标数据集 WebLogo-2M [20] 还多出了数千倍的不同 Logo。

与常用的自然图像数据集 (如 ImageNet [16]、CIFAR-10 [10] 和 LSUN [26])、人脸数据集 (如 CelebA [12]) 和相对容易建模的 MNIST 手写数字 [11] 相比，Logo 数据集是:(1) 多模态的，对生成模型具有挑战性;(2) 应用性质的，现实世界对合成的、独特的 Logo 有明显的需求;(3) 很难确定标签的，Logo 的视觉外观中很少有明确的属性，这确保了大型 Logo 数据集的多样性。

Dataset	Logos	Images
FlickLogos-27 [9]	27	1080
FlickLogos-32 [15]	32	8240
BelgaLogos [8]	37	10000
LOGO-Net [6]	160	73414
WebLogo-2M [20]	194	1867177
LLD-icon(ours)	486377	486377
LLD-logo(ours)	122920	122920
LLD(ours)	486377+	609297

表 1. 大型公共 Logo 数据集，后三行为该课题构建的 LLD

2.2 多模态数据上训练 GANs 模型

该课题提出了一种稳定 GANs 训练的方法，并通过聚类 (a) 在相同数据上训练的自编码器潜在空间中，或者通过聚类 (b) 在 ImageNet 上训练的 ResNet 分类器 CNN 特征空间中获

得对发生器输出的额外控制。通过这两种方法，都能够生成语义上有意义的聚类，从而改进 GANs 的训练。在下一节的方法介绍中，将会回顾研究中使用的 GANs 架构，并且描述基于 Autoencoder 潜空间和 ResNet 特征的聚类方法。

3 本文方法

本文的主要复现工作集中在该课题的多模态数据上稳定训练 GANs 的方法上，而且这也是该课题的主要工作和贡献。此章节主要对聚类 GANs 训练、聚类方法以及使用的 GANs 架构进行介绍。

3.1 GAN 架构

该课题的生成模型是基于 Radford 等人提出的 [14] 的深度卷积生成对抗网络 (DCGAN) 和 Gulrajani 等人 [4] 提出的带有梯度惩罚的改进 Wasserstein GAN (iWGAN)。

对于该课题的 DCGAN 实验，使用了 Taehoon Kim 的 TensorFlow 实现。该课题专门在低分辨率的 LLD-icon 子集上训练 DCGAN，有关这种低分辨率的子集，如果不使用该课题所提的聚类方法，训练 DCGAN 的过程将会不稳定，所有的 DCGAN 实验将使用下一节中介绍的输入模糊。对于该课题的 iWGAN 实验都是基于 Gulrajani 等人 [4] 提出的官方 TensorFlow 存储库，并且使用了 Gulrajani 等人 [4] 设置的默认参数。

3.2 聚类方法

该课题发现，对于分辨率高于 10×10 的图标数据集 (LLD-icon)，DCGAN 是不稳定的，并且可以通过引入本节所述的合成标签来稳定它。除了稳定 GAN 训练之外，在第 3.4 节中，该课题能够在 CIFAR-10 和 ImageNet-Small [22] 上使用 iWGAN 和下面描述 RC 聚类生成的合成标签，在 Inception 分数 (由 Salimans 等人提出 [19]) 上实现显著改进。基于此，该课题提出了两种不同的方法在训练数据上生成合成数据标签。

自动编码器聚类（AutoEncoder Clustering，AE），在训练了一个与 GANs 结构相似的自编码器之后，接着在训练数据上对该自编码器的潜在空间 z 中的图像进行聚类。由于该潜在空间与学习到的高级 AE 特征直接相关，因此生成的聚类在语义上有意义，并且易于被 GAN 拾取 (由于类似的网络架构)。图 2给出了该方法的详细信息。

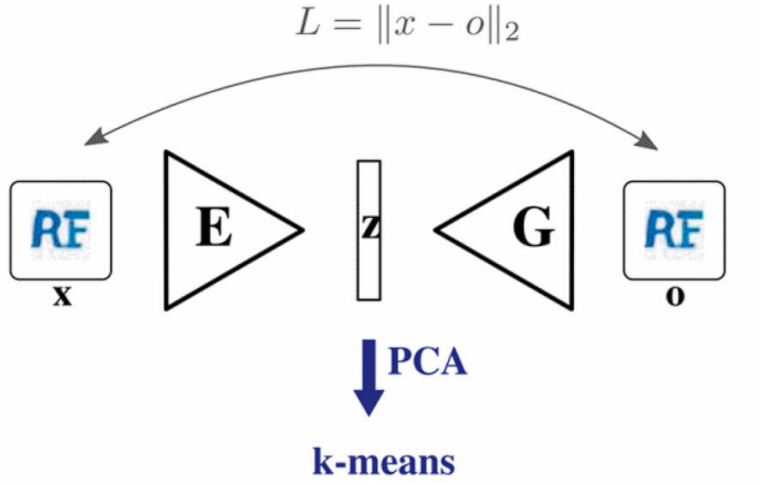

图 2. 用于 AE 聚类的自动编码器，G 是 GAN 的发生器，E 是 GAN 的判别器 D 的组成部分，具有更高的输出，以匹配潜在空间 z 的维度。自动编码器使用 L2 损失函数进行训练

ResNet 分类器聚类 (ResNet Classifier Clustering, RC)，对于这种聚类方法，该课题利用了从 ImageNet 分类器中学习到的特征，即 He 等人 [5][13] 的 ResNet-50。与 AE 聚类一样，在使用 (minibatch) k-means 聚类特征之前，使用 PCA 从 ResNet 网络的最终池化层降低特征向量的维数。

ResNet 分类器聚类方法在 CIFAR-10 上给出了相当好的结果 (考虑到它与 ImageNet 的相似性，这并不奇怪)，并且对于不同的图像数据 (如 LLD 数据集)，ResNet 分类器聚类仍然表现的很好。考虑到该课题使用以监督方式训练的分类器，不需要对训练 GAN 本身的数据进行任何标注，因此将这种方法称为半监督方法。

3.3 条件 GAN 训练方法

层条件 GAN（Layer Conditional GAN，LC）模型，顾名思义，就是将每个训练样本的聚类标签传输到发生器和判别器的所有卷积层和线性层。对于线性层，标签只是作为一个单热向量附加到输入上。对于卷积层，标签被投影到“单热特征图”上，通道数量与簇的数量一样多，其中与簇号对应的一个被 1 填充，而其余的都是 0。这些附加的特征映射被附加到每个卷积层的输入，这样每一层都可以直接访问标签信息。在该课题的 iWGAN 模型中使用 DCGAN 图 3和 ResNet 图 4体现出了这一点。即使标签提供给每一层，也没有明确的机制强制网络使用这些信息。如果标签是随机的或无意义的，它们可以被网络忽略。一旦 GANs 判别器开始为每个聚类调整其标准，层条件 GAN 模型就会强制发生器输出每个类不同要求的图像。该课题的实验表明，视觉上有意义的聚类总是能够被模型识别。

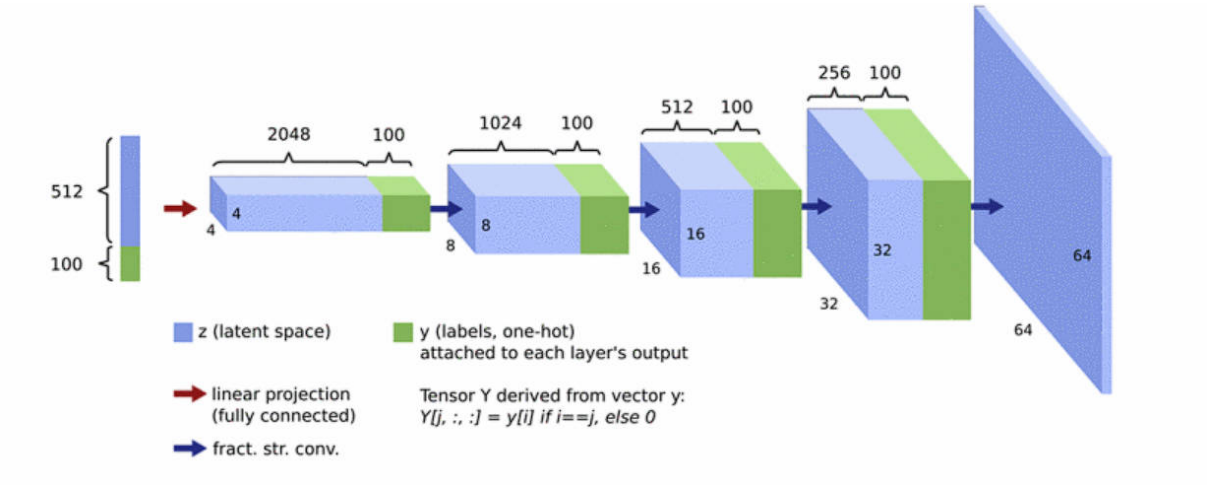

图 3. 发生器网络用于层条件 DCGAN (DCGAN-LC),100 个标签 y 作为一个单热向量附加到潜在向量上，标签 y 还被投影到一组特征映射上，最后将这些特征映射作为每个卷积层的输入

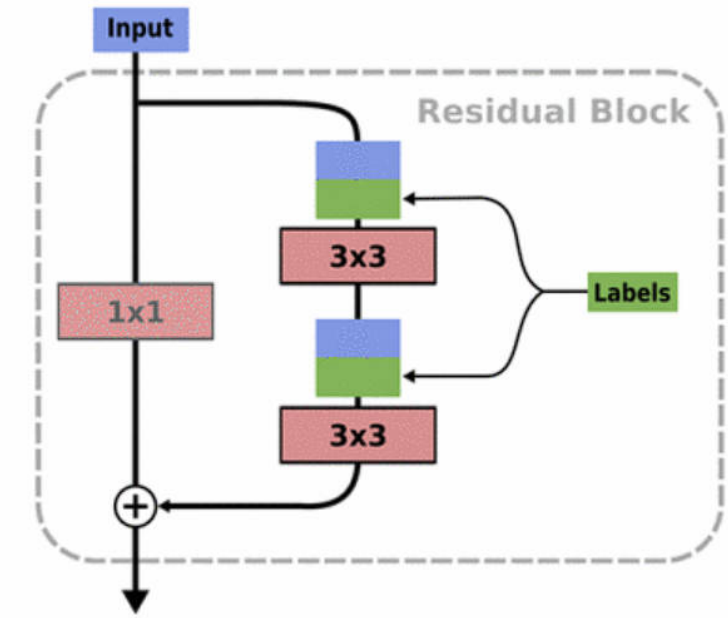

图 4. 在 iWGAN-LC 中使用的层条件剩余块，标签信息与图 3所示相同的方式附加到卷积层的输入

辅助分类器 GAN（Auxiliary Classifier GAN，AC），对于 iWGAN，该课题使用了由 Odena 等人 [13] 提出的辅助分类器，并由 Glurajani 等人 [4] 实现。这种方法不容易在聚类之间进行插值，因此从应用程序的角度来看，辅助分类器 GAN 会稍微受到一些限制，但避免了向卷积层添加参数，这通常会导致网络的参数更少。iWGAN-AC 是我们在数据集 CIFAR-10 中选择的方法，因为它提供了最高的 Inception 分数。

高斯模糊算法（Gaussina Blur），在该课题的实验中，模糊输入图像有利于帮助 GAN 在训练期间保持稳定，基于 Susmelj 等人做过类似的模糊处理 [21]，该课题最终对呈现给判别器的所有图像 (训练数据以及来自发生器的样本) 应用高斯模糊。该方法如图 5所示。

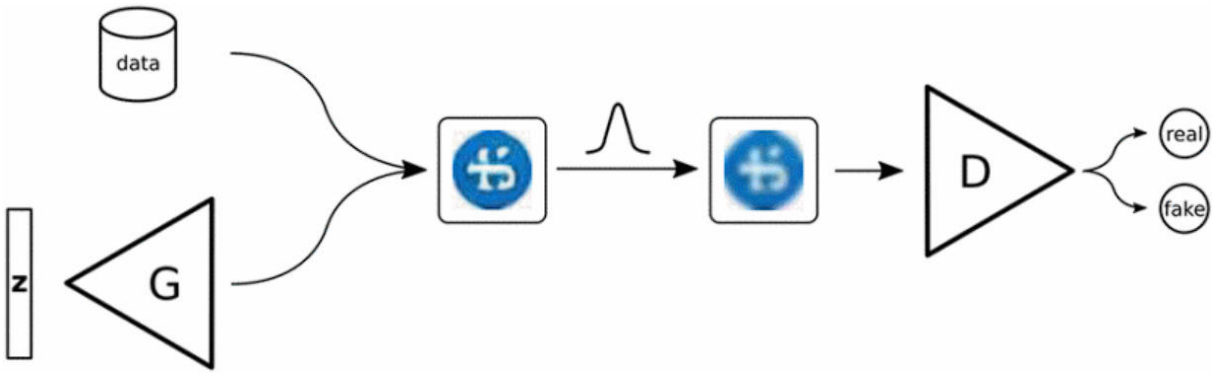

图 5. 模糊鉴别器处理之后输入 GAN，原始图像和生成图像都使用固定强度的高斯滤波器进行模糊处理

4 复现细节

复现的论文已有源码，由于源码代码量较大，主要展示源码的模型部分和自己稍加修改的部分。详细的代码实现参考：基于聚类的生成式对抗网络合成和操纵 Logo 的源码实现，训练数据集下载参考：LLD-模型预训练的大型 Logo 数据集下载 [17]。

4.1 模型代码

该代码属于 DCGAN 模型的发生器和鉴别器两个重要部分。生成器主要用于生成图像，它接受一个随机向量 z 作为输入，并根据是否给定标签 y 来生成相应的图像。如果给定了标签 y，生成的图像将与 y 有关；如果没有给定标签 y，生成的图像将是随机的。判别器主要根据是否给定了标签 y 来决定生成的图像是否会考虑这个条件，如果给定了标签 y，将标签 y 转换为适当的形状，并将其与图像连接起来，然后，通过一系列卷积层和批量归一化层对图像进行处理。最后，将处理后的图像展平并连接到一个线性层，以输出一个标量值。如果没有给定标签 y，直接对图像进行一系列卷积层和批量归一化层的处理，然后将处理后的图像展平并连接到一个线性层，以输出一个标量值。

```
#判别器
def discriminator(self, image, y=None, reuse=False, sampler=False):
"""Defines the discriminator CNN architecture"""
with tf.variable_scope('discriminator') as scope:
    train = True
    if sampler:
        train = False
        reuse = True
    if reuse:
        scope.reuse_variables()
    if y is not None:
        yb = tf.reshape(y, [self.batch_size, 1, 1, self.y_dim])
        x = conv_cond_concat(image, yb)
```

```
14      if y is None:
15          h0 = lrelu(conv2d(image, self.df_dim, k_h=self.df_size, k_w = self.
                df_size, name='d_h0_conv'))
16          h1 = lrelu(self.d_bn1(conv2d(h0, self.df_dim * 2, k_h = self.
                df_size, k_w=self.df_size, name='d_h1_conv'),
17          train=train))
18
19          h2 = lrelu(self.d_bn2(conv2d(h1, self.df_dim * 4, k_h = self.
                df_size, k_w=self.df_size, name='d_h2_conv'),
20          train=train))
21
22          h3 = lrelu(self.d_bn3(conv2d(h2, self.df_dim * 8, k_h = self.
                df_size, k_w=self.df_size, name='d_h3_conv'),
23          train=train))
24
25          h4 = linear(tf.reshape(h3, [self.batch_size, -1]), 1, 'd_h3_lin')
26      else:
27          yb = tf.reshape(y, [self.batch_size, 1, 1, self.y_dim])
28
29          h0 = lrelu(conv2d(conv_cond_concat(image, yb), self.df_dim + self.
                y_dim, k_h=self.df_size, k_w = self.df_size, name ='d_h0_conv'))
30
31          h1 = lrelu(self.d_bn1(conv2d(conv_cond_concat(h0, yb), self.df_dim
                * 2 + self.y_dim, k_h=self.df_size, k_w = self.df_size, name='
                d_h1_conv'), train=train))
32
33          h2 = lrelu(self.d_bn2(conv2d(conv_cond_concat(h1, yb), self.df_dim
                * 4 + self.y_dim, k_h = self.df_size, k_w = self.df_size, name='
                d_h2_conv'), train=train))
34
35          h3 = lrelu(self.d_bn3(conv2d(conv_cond_concat(h2, yb), self.df_dim
                * 8 + self.y_dim, k_h=self.df_size, k_w = self.df_size, name='
                d_h3_conv'), train=train))
36
37          h3_flat = tf.reshape(h3, [self.batch_size, -1])
38
39          h4 = linear(concat([h3_flat, y], 1), 1, 'd_h3_lin')
40
41      return tf.nn.sigmoid(h4), h4
42  #发生器
```

```
43 def generator(self, z, y=None, sampler=False):
44 with tf.variable_scope('generator') as scope:
45     train = True
46     if sampler:
47         train = False
48         scope.reuse_variables()
49     s_h, s_w = self.output_height, self.output_width
50 
51     s_h2, s_w2 = conv_out_size_same(s_h, 2), conv_out_size_same(s_w, 2)
52 
53     s_h4, s_w4 = conv_out_size_same(s_h2, 2), conv_out_size_same(s_w2, 2)
54 
55     s_h8, s_w8 = conv_out_size_same(s_h4, 2), conv_out_size_same(s_w4, 2)
56 
57     s_h16, s_w16 = conv_out_size_same(s_h8, 2), conv_out_size_same(s_w8,
        2)
58 
59     if y is not None:
60         z = concat([z, y], 1)
61         z_, w_lin, b_lin = linear(z, self.gf_dim * 8 * s_h16 * s_w16, '
            g_h0_lin', with_w=True)
62         w_y = tf.slice(w_lin, [self.z_dim, 0], [self.y_dim, self.gf_dim * 8
             * s_h16 * s_w16])
63         self.w_y_sum = tf.summary.histogram('g_label_weigths_histogram',
            w_y)
64     else:
65         z_ = linear(z, self.gf_dim * 8 * s_h16 * s_w16, 'g_h0_lin')
66 
67 
68     h0 = tf.reshape(z_, [-1, s_h16, s_w16, self.gf_dim * 8])
69 
70     h0 = tf.nn.relu(self.g_bn0(h0, train=train))
71 
72     if y is None:
73         print('no y given')
74 
75         h1 = deconv2d(h0, [self.batch_size, s_h8, s_w8, self.gf_dim * 4],
            k_h=self.gf_size, k_w=self.df_size, name='g_h1')
76 
77         h1 = tf.nn.relu(self.g_bn1(h1, train=train))
```

```
78
79          h2 = deconv2d(h1, [self.batch_size, s_h4, s_w4, self.gf_dim * 2],
                k_h=self.gf_size, k_w=self.df_size, name='g_h2')
80
81          h2 = tf.nn.relu(self.g_bn2(h2, train=train))
82
83          h3 = deconv2d(h2, [self.batch_size, s_h2, s_w2, self.gf_dim],
84          k_h=self.gf_size, k_w=self.df_size, name='g_h3')
85
86          h3 = tf.nn.relu(self.g_bn3(h3, train=train))
87
88          h4 = deconv2d(h3, [self.batch_size, s_h, s_w, self.c_dim],
89
90          k_h=self.gf_size, k_w=self.df_size, name='g_h4')
91      else:
92          print('y given')
93          yb = tf.reshape(y, [self.batch_size, 1, 1, self.y_dim])
94
95          h1 = deconv2d(conv_cond_concat(h0, yb), [self.batch_size, s_h8,
                s_w8, self.gf_dim * 4 + self.y_dim], k_h=self.gf_size, k_w=self.
                df_size, name='g_h1')
96
97          h1 = tf.nn.relu(self.g_bn1(h1, train=train))
98
99          h2 = deconv2d(conv_cond_concat(h1, yb), [self.batch_size, s_h4,
                s_w4, self.gf_dim * 2 + self.y_dim], k_h=self.gf_size, k_w=self.
                df_size, name='g_h2')
100
101         h2 = tf.nn.relu(self.g_bn2(h2, train=train))
102
103         h3 = deconv2d(conv_cond_concat(h2, yb), [self.batch_size, s_h2,
                s_w2, self.gf_dim + self.y_dim], k_h=self.gf_size, k_w=self.
                df_size, name='g_h3')
104
105         h3 = tf.nn.relu(self.g_bn3(h3, train=train))
106
107         h4 = deconv2d(conv_cond_concat(h3, yb), [self.batch_size, s_h, s_w,
                 self.c_dim], k_h=self.gf_size, k_w=self.df_size, name='g_h4')
108     return tf.nn.tanh(h4)
```

4.2 实验环境搭建

实验环境设置如下：

- Python 3.8.10
- Pytorch 1.7.1
- tensorflow 2.13.0
- h5py 3.10.0

实验运行硬件环境为一张 Tian X Pascal。由于设备资源有限，只在部分的训练数据集 LLD 上训练了 500 个伦次。

5 实验结果分析

根据第 3 章所介绍的相关方法，该课题基于 CIFAR-10 和 ImageNet-small 两个数据集，通过对比不同的 GAN 模型训练过程，得到了 MS-SSIM [23] 的 Inception 分数 [19] 和 Diversity 分数，这是在 [13] 中提出的，基于 50000 张随机生成的图像。在表 2中，我们总结了 LC(层条件生成式对抗网络) 和 AC（辅助分类器生成式对抗网络）条件模式下有监督 (使用 CIFAR 标签) 和无监督设置的不同配置的结果。

Method	Clusters	Inception score	Diversity(MS-SSIM)
Infusion training [1]		4.62±0.06	
ALI [3](from [24])		5.34±0.05	
Impr.GAN(-L+HA) [19]		6.86±0.06	
EGAN-Ent-VI [2]		7.07±0.10	
DFM [24]		7.72±0.13	
iWGAN [4]		7.86±0.07	
iWGAN		7.853±0.072	0.0504±0.0017
iWGAN-LC with AE clustering	32	7.300±0.072	0.0507±0.0016
iWGAN-AC with AE clustering	32	7.885±0.083	0.0504±0.0014
iWGAN-LC with RC clustering	32	7.831±0.072	0.0491±0.0015
iWGAN-LC with RC clustering	128	7.799±0.030	0.0491±0.0015
iWGAN-AC with RC clustering	10	8.433±0.068	0.0505±0.0016
iWGAN-AC with RC clustering	32	**8.673±0.075**	0.0500±0.0016
iWGAN-AC with RC clustering	128	**8.625±0.109**	**0.0465±0.0015**
iWGAN-LC		7.710±0.084	0.0510±0.0013
Impr.GAN [19]		8.09±0.07	
iWGAN-AC [4]		8 .42±0.10	
AC-GAN [13]		8.25±0.07	
SGAN [7]		8.59±0.12	
CIFAR-10 (original data)		11.237±0.116	0.0485±0.0016

表 2. 各个模型和方法在 CIFAR-10 上 Inception 和 Diversity 的评分比较，无监督方法不使用 CIFAR-10 类标签，其中第一部分为无监督方法，第二部分为半监督方法，第三部分为监督方法

Method		Inception score
iWGAN	unconditional	10.11±0.20
iWGAN-AC	128 RC clusters	14.42±0.21
ImageNet-small	(original data)	75.29±1.40

表 3. iWGAN 在 ImageNet-small 上的 Inception 评分比较

在 CIFAR-10 数据集上，该课题使用 iWGAN-AC 和 32 个 RC 聚类获得的最佳 Inception 分数为 8.67，显著高于 Salimans 等人 [19] 使用他们改进的 GAN 方法获得的 8.09，这是文献中无监督方法显示的最佳分数。此外，该课题使用 RC 聚类提供的纯合成标签获得的最佳结果与 Huang 等人 [7] 的堆叠 gan 方法的 8.59 分相当，后者是监督方法报告的最佳分数。对

于 ImageNet-small 来说，与不使用标签相比，使用合成标签也有非常显著的改进。虽然这些结果可以指出使用数据聚类时，输出图像质量可以得到改善，但该课题认为更高的分数归因于 AC-GAN（辅助分类器生成式对抗网络）强制生成的图像可以很容易地分类到所提供的聚类中，这反过来可以提高基于分类器的 Inception 分数。

除了 Inception 和 Diversity 评分外，该课题使用 CORNIA 来测量图像质量，CORNIA 是 Ye 和 Doermann [25] 提出的一种鲁棒的无参考图像质量评估方法。在 CIFAR-10 和 LLD-icon 上，该课题提出的生成模型获得的 CORNIA 分数与每个数据集的原始图像相当。这一结果与 [21] 的研究结果一致，在 GAN 收敛时，所研究的 GAN 在 CORNIA 分数方面也趋于数据集的原始图像。

6 总结与展望

在该课题中，作者们通过生成模型的综合和操作来解决 Logo 设计在打草稿中需要花费大量时间的问题:

- 作者们引入了一个从互联网上抓取的大型徽标数据集 (LLD)，该数据集的 Logo 数量比互联网存在的 Logo 数据集多几个数量级。
- 为了使多模态数据在 GAN 上训练时的稳定性，作者们提出了聚类 GAN，即通过聚类获得的合成标签作为条件的 GAN。此外，作者们在自编码器的潜在空间或 ResNet 分类器的 CNN 特征空间和条件 DCGAN 中进行聚类，并利用辅助分类器或层条件模型改进 WGAN。
- 作者们在 CIFAR-10 和 ImageNet 上定量验证了他们所提出的聚类 GAN 方法。

对于该课题所提出的使用聚类生成式对抗网络实现 Logo 的合成和操作，我认为值得进一步改进的地方有，作者可以进一步优化 CGAN 的模型结构和训练策略，以提高其生成和操作标志的效率和质量，此外，作者还可以探索将 CGAN 应用于其他类型的图像生成和操作任务，例如人脸合成和风格迁移等。

参考文献

[1] Florian Bordes, Sina Honari, and Pascal Vincent. Learning to generate samples from noise through infusion training. *arXiv preprint arXiv:1703.06975*, 2017.

[2] Zihang Dai, Amjad Almahairi, Philip Bachman, Eduard Hovy, and Aaron Courville. Calibrating energy-based generative adversarial networks. *arXiv preprint arXiv:1702.01691*, 2017.

[3] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[4] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Steven CH Hoi, Xiongwei Wu, Hantang Liu, Yue Wu, Huiqiong Wang, Hui Xue, and Qiang Wu. Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. *arXiv preprint arXiv:1511.02462*, 2015.

[7] Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5077–5086, 2017.

[8] Alexis Joly and Olivier Buisson. Logo retrieval with a contrario visual query expansion. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 581–584, 2009.

[9] Yannis Kalantidis, Lluis Garcia Pueyo, Michele Trevisiol, Roelof van Zwol, and Yannis Avrithis. Scalable triangulation-based logo recognition. In *Proceedings of the 1st ACM international conference on multimedia retrieval*, pages 1–7, 2011.

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[13] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[14] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[15] Stefan Romberg, Lluis Garcia Pueyo, Rainer Lienhart, and Roelof Van Zwol. Scalable logo recognition in real-world images. In *Proceedings of the 1st ACM international conference on multimedia retrieval*, pages 1–8, 2011.

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[17] Alexander Sage, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Lld - large logo dataset - version 0.1. https://data.vision.ee.ethz.ch/cvl/lld, 2017.

[18] Hichem Sahbi, Lamberto Ballan, Giuseppe Serra, and Alberto Del Bimbo. Context-dependent logo matching and recognition. *IEEE Transactions on Image Processing*, 22(3):1018–1031, 2012.

[19] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[20] Hang Su, Shaogang Gong, and Xiatian Zhu. Weblogo-2m: Scalable logo detection by deep learning from the web. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 270–279, 2017.

[21] Igor Susmelj, Eirikur Agustsson, and Radu Timofte. Abc-gan: Adaptive blur and control for improved training stability of generative adversarial networks. In *International Conference on Machine Learning (ICML 2017) Workshop on Implicit Models*, volume 5, 2017.

[22] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[23] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[24] David Warde-Farley and Yoshua Bengio. Improving generative adversarial networks with denoising feature matching. In *International conference on learning representations*, 2016.

[25] Peng Ye and David Doermann. No-reference image quality assessment using visual codebooks. *IEEE Transactions on Image Processing*, 21(7):3129–3138, 2012.

[26] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[27] Guangyu Zhu and David Doermann. Automatic document logo detection. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 864–868. IEEE, 2007.