用更少的标签生成高保真的图像

Mario Lucic [*1] Michael Tschannen [*2] Marvin Ritter [*1] Xiaohua Zhai [1] Olivier Bachem [1] Sylvain Gelly [1]

摘 要

深度生成模型正在成为现代机器学习的重要组成部分。 最近关于条件生成对抗网络的研究表明，学习复杂的、高维的自然图像分布是可以实现的。虽然最新模型能够以高分辨率生成高保真、多样化的自然图像，但它们依赖于大量标记数据。 在这项工作中，我们展示了如何从最近关于自适应学习和半监督学习的研究工作中获益，以便在无监督的 ImageNet 上合成以及有条件数据集中超越业界最高指标（SOTA）。 特别是，我们提出的方法仅使用 10%的标签时能够匹配 ImageNet 上当前最先进的条件模型 Big-GAN 的样本质量（通过 FID 测量），并且在使用 20%的标签时超过它。

1. 介绍

深度生成模型由于具有学习复杂高维分布的能力而受到极大关注，例如自然图像的分布 (Zhang et al., 2018; Brock et al., 2019), 视频 (Kalchbrenner et al., 2017), 以及声频 (Van Den Oord et al., 2016)。最近的进展是由大规模模型的可扩展训练推动的 (Brock et al., 2019; Menick & Kalchbrenner, 2019), 架构修改 (Zhang et al., 2018; Chen et al., 2019a; Karras et al., 2018), 以及正则化技巧(Miyato et al., 2018)。

高保真自然图像生成（通常在 ImageNet 上训练）取决于可以访问大量标记数据。 这并不令人惊讶，因为标签会将丰富的辅助信息引入训练过程，有效地将极具挑战性的图像生成任务划分为具有语义意义的子任务。

[*]同等贡献: [1]Google Brain, Zurich, Switzerland [2]ETH Zurich, Zurich, Switzerland. Correspondence to: Mario Lucic <lucic@google.com>, Michael Tschannen <mi.tschannen@gmail.com>, Marvin Ritter <marvinritter@google.com>.

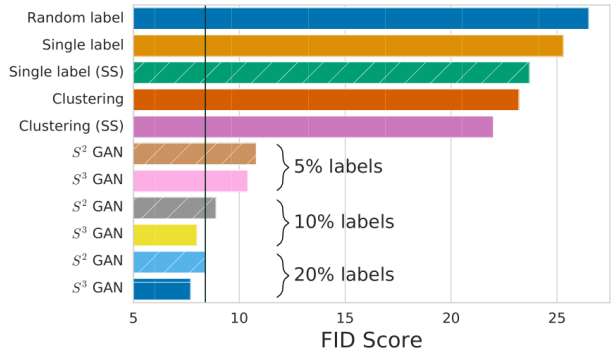


图 1.基线的 FID 和提议的方法。 垂直线表示使用所有标记数据的基线（BigGAN）。 我们提出的方法（S^3 GAN）在仅使用 10%的标记数据时能够匹配最先进的技术，同时使用 20%时会超过它。

然而，这种对大量标记数据的依赖与大多数数据未标记的事实不一致，并且标记本身通常是昂贵且容易出错的。尽管最近在无监督图像生成方面取得了进展，但在样本质量方面，条件模型和非监督模型之间的差距是显著的。

在这项工作中，我们迈出了重要的一步，使用生成对抗网络（GAN）缩小有条件和无监督生成高保真图像之间的差距。 我们利用两个简单而强大的概念：

(i) 自我监督学习：可以通过自我监督来学习用于训练数据的语义特征提取器，然后可以使用所得到的特征表示来指导 GAN 训练过程。

(ii) 半监督学习：可以从标记的训练图像的一小部分推断出整个训练集的标签，并且推断的标签可以用作 GAN 训练的条件信息。

我们的贡献 在这项工作中，我们

1. 提出并研究各种方法，以减少或完全忽略自然图像生成任务的完全真实标签信息，
2. 在 Imagenet 上实现无监督学习生成的新 State Of The Art(下称 SOTA)，仅使用 10％的标签匹配 128×128 IMAGENET 上的 SOTA，并仅使用 20%的标签设置得到新的 SOTA（通过 FID 测量），以及
3. 开源所有用于实验的代码：github.com/google/compare_gan.

2. 背景和相关工作

IMAGENET 上的高保真 GAN 除了 BigGAN（Brock 等，2019）之外，只有少数先前的方法具有将 GAN 扩展到 ImageNet 的规模，其中大多数依赖于使用标签的类条件生成。最早的尝试之一是具有辅助分类器（AC-GAN）的 GAN（Odena 等人，2017），其将具有潜在编码的 one-hot 编码标签信息馈送到生成器并且为判别器配备顶部辅助以预测图像输入是真的还是假的。最近的方法依赖于判别器中的标签投影层，基本上导致每类真/假分类（Miyato & Koyama，2018）和生成器中的自我关注（Zhang 等，2018）。两种方法都使用调制批量标准化（modulated batch normalization）（De Vries 等，2017）来向生成器提供标签信息。在无监督领域，陈等人（2019b）表明，添加到判别器的辅助旋转损失（rotation loss）对训练具有稳定作用。最后，适当的梯度正则化可以在不使用标签的情况下将 MMD-GAN 扩展到 ImageNet（Arbel 等，2018）。

半监督的 GAN 最近的一些工作利用 GAN 进行分类器的半监督学习。两个工作：Salimans 等人（2016）和 Odena（2016）训练一个判别器，将其输入分类为 K + 1 类：真实图像的 K 图像类和生成图像的一个类。同样，Springenberg（2016）将标准 GAN 目标扩展到 K 类。 Li 等人也考虑了这种方法（2017）应用单独的判别器和分类器模型。其他方法结合推理模型来预测缺失标签（Deng et al，2017）或利用匹配半监督学习的联合分布（标签和数据）（Gan et al，2017）。我们强调，这一系列工作侧重于从几个标签中训练分类器，而不是使用少量标签来提高生成模型的质量。据我们所知，Li 等人（2017），Deng 等人（2017）以及 Sricharan 等人（2017）报道了通过部分标签信息改善样品质量，所有这些都只考虑来自受限域的低分辨率数据集。

自我监督学习 自我监督学习方法采用无标签辅助任务来学习数据的语义特征表示。这种方法成功地应用于不同的数据模态，如图像（Doersch 等，2015; Caron 等，2018），视频（Agrawal 等，2015; Lee 等，2017）和机器人技术（Jang 等，2018; Pinto & Gupta，2016）。IMAGENET 目前最先进的方法是由 Gidaris 等人提出的（2018）将旋转训练图像的旋转角度预测为辅助任务。这种简单的自我监督方法产生了对下游图像分类任务有用的表示。其他形式的自我监督包括

图 2.顶行：来自完全监督的当前最先进模型 BigGAN 的 128 × 128 样本。 底行：样本来自于提出的S^3 GAN，它使用 FID 和 IS 匹配 BigGAN，仅使用 10%的完全真实标签。

预测给定图像的不相交图像块的相对位置（Doersch 等，2015; Mundhenk 等，2018）或估计在规则网格上随机交换的图像块的置换（Noroozi & Favaro，2016）。 Kolesnikov 等人提供了一种利用现代神经结构进行自我监督学习的研究（2019）。

3. 降低标签数据的需求

简而言之，我们将提供推断的，而不是为判别器提供真实图像的手工注释的完全真实标签。为了获得这些标签，我们将利用自我和半监督学习的最新进展。在详细介绍这些方法之前，我们首先讨论如何在最先进的 GAN 中使用标签信息。以下说明假定您熟悉 GAN 框架的基础知识（Goodfellow 等，2014）。

合并标签 为了向判别器提供标签信息，我们采用了 Miyato & Koyama（2018）提出的线性投影层。为了使展览自成一体，我们将简要回顾一下主要观点。在“vanilla”（无条件）GAN 中，判别器 D 学习预测其输入 x 处的图像是真实的还是由生成器 G 生成的。我们将判别器分解为学习到的判别器表示，$\tilde{\mathrm{D}}$，其被馈送到线性分类器：$\mathrm{c}_{r/f}$中，即鉴别器由$c_{\mathrm{r/f}}(\tilde{D}(x))$给出。在投影判别器中，学习与表示相同维度的每个类的嵌入$\tilde{D}(x)$。然后，对于给定的图像，标签输入 x，y 关于样本是真实的还是生成是基于两个组成部分决定：（a）表示$\tilde{D}(x)$本身是否与实际数据一致，以及（b）表示$\tilde{D}(x)$是否与类别 y 中的实际数据一致。更正式地，判别器采用$D(x,y) = c_{\mathrm{r/f}}(\tilde{D}(x)) + P(\tilde{D}(x), y)$的形式，其中$P(\tilde{x}, y) = \tilde{x}^{\top} W y$是线性投影层应用于特征向量 $\tilde{\mathrm{x}}$ 和 one-hot 编码标签 y 作为

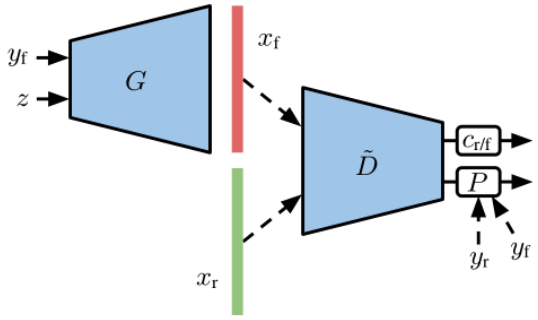


图 3.带有投影判别器的条件 GAN。 判别器试图通过组合无条件分类器$c_{r/f}$和通过投影层 P 实现的类条件分类器来从表示$\tilde{D}$中预测实际图像 x_r（具有标签 y_r）或生成图像 x_f（具有标签 y_f）是否在其输入处。 这种形式的调理用于 BIGGAN。 向外箭头传给损失函数。

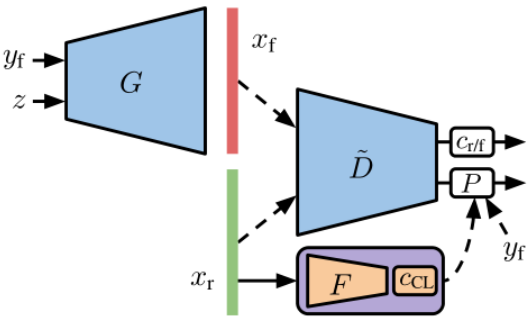


图 4.聚类：基于聚类通过求解自监督任务获得表示的无监督方法。 F 对应于通过自我监督学习的特征提取器，c_{CL}是聚类分配函数。 在预训练步骤中在真实训练图像上学习 F 和c_{CL}之后，我们通过将标签推断为$\hat{y}_{\mathrm{CL}} = c_{\mathrm{CL}}(F(x))$来进行条件 GAN 训练。

输入。对于生成器，标签信息 y 通过类条件 BatchNorm（Dumoulin 等，2017; De Vries 等，2017）进行了补充。带有投影判别器的条件 GAN 如图 3 所示。我们继续描述预训练和协同训练的方法，分别推断 3.1 和 3.2 节中 GAN 训练的标签。

3.1 预先训练的方法

基于无监督聚类的方法 我们首先使用最先进的自我监督方法（Gidaris et al，2018; Kolesnikov et al，2019）学习真实训练数据的表示，对此表示进行聚类，并使用聚类项作为替代的标签。 继 Gidaris 等人之后（2018）我们通过最小化以下自我监督损失来学习特征提取器 F（通常是卷积神经网络）

$$\mathcal{L}_{\mathrm{R}} = -\frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\log p(c_{\mathrm{R}}(F(x^r)) = r)], \quad (1)$$

其中 R 是 4 个旋转度的集合 $\{0°, 90°, 180°, 270°\}$，x^r是由 r 旋转的图像 x，并且c_R是预测旋转度 r 的线性分类器。在学习了特征提取器 F 之后，我们对训练图像的表示应用了小批量 k-Means 聚类（Sculley，2010）。 最后，给定聚类分配函数 $\hat{y}_{\mathrm{CL}} = c_{\mathrm{CL}}(F(x))$我们使用铰链损失训练 GAN，或者最小化判别器损失$\mathcal{L}_D$和生成器损失$\mathcal{L}_G$，即

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\min(0, -1 + D(x, c_{\mathrm{CL}}(F(x))))]\\ &-\mathbb{E}_{(z,y)\sim\hat{p}(z,y)}[\min(0, -1 - D(G(z,y),y))]\\ \mathcal{L}_G = &-\mathbb{E}_{(z,y)\sim\hat{p}(z,y)}[D(G(z,y),y)],\end{aligned}$$

其中 $\hat{p}(z,y) = p(z)\hat{p}(y)$ 是先验分布，满足 $p(z) = \mathcal{N}(0, I)$，并且 $\hat{p}(y)$是经验分布关于训练集上的聚类标签 $c_{\mathrm{CL}}(F(x))$。 我们将这种方法称为聚类，并在图 4 中对其进行说明。

半监督方法 虽然半监督学习是一个活跃的研究领域，并且已经提出了各种各样的算法，但我们遵循 Beyer 等人的观点（2019）并简单地将前一段中描述的自我监督方法扩展为半监督损失，这确保了两种方法在模型容量和计算成本方面具有可比性。 假设我们为训练数据的子集提供了标签，我们尝试通过自我监督学习一个好的特征表示，同时在如此获得的表示上训练一个好的线性分类器（使用提供的标签，见页底 1）。更多正式地，我们将损失降至最低

$$\begin{aligned}\mathcal{L}_{\mathrm{S^2L}} = -\frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\Big\{&\mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\log p(c_{\mathrm{R}}(F(x^r)) = r)]\\ &+\gamma\mathbb{E}_{(x,y)\sim p_{\mathrm{data}}(x,y)}[\log p(c_{\mathrm{S^2L}}(F(x^r)) = y)]\Big\},\end{aligned} \quad (2)$$

其中 c_R 和 $c_{\mathrm{S^2L}}$ 分别是预测旋转角度 r 和标签 y 的线性分类器，并且γ > 0 平衡损失项。 （2）中的第一项对应于（1）的自我监督损失，而第二项对应于（半监督的）交叉熵损失。 在训练期间，后者的期望被标记的训练样本子集的经验平均值所取代，而前者被设置为整个训练集的经验平均值（整个论文都遵循该惯例）。 在我们获得 F 和 $c_{\mathrm{S^2L}}$后，我们继续进行 GAN 训练

[1] 请注意，更简单的方法是首先通过自我监督和随后的线性分类器来学习表示，但我们观察到同时学习表示和分类器会产生更好的结果。

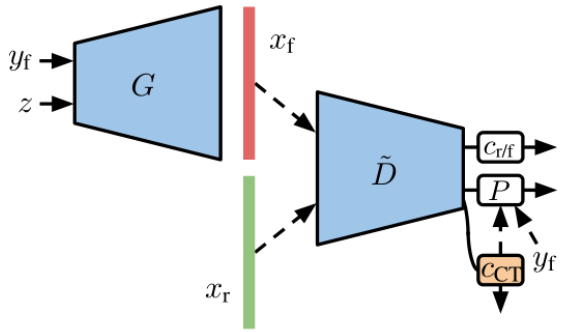


图 5. S²GAN-CO：在 GAN 训练期间，我们在判别器表示上学习辅助分类器 c_{CT}，基于 $\tilde{D}$ 标记的实例预测未标记的标签。 这避免了在 GAN 训练之前训练特征提取器 F 和分类器 c_{S^2L}，如在S²GAN 中那样。

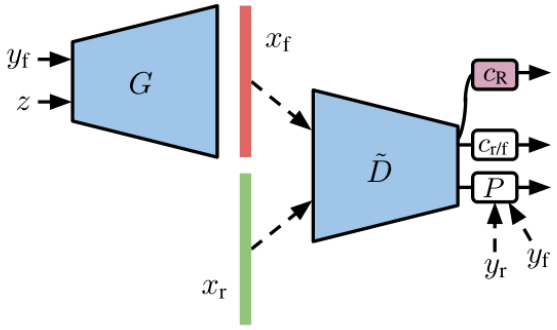


图 6. 在 GAN 训练期间通过旋转预测进行自我监督。 除了预测其输入处的图像是真实的还是生成的之外，还训练判别器以通过辅助线性分类器 c_R预测真实和伪造图像的旋转。 Chen 等人成功应用了这种方法（2019b）稳定 GAN 训练。 在这里，我们将它与我们的预训练和协同训练方法相结合，用预测的方法替换完全真实标签。

将真实图片标记为$\hat{y}_{S^2L} = c_{S^2L}(F(x))$。 特别地，除了我们使用通过最小化（2）获得的 c_{S^2L}和 F 之外，我们选择性地最小化与聚类相同的生成器和判别器损失：

$$\mathcal{L}_D = -\mathbb{E}_{x\sim p_{\text{data}}(x)}[\min(0, -1 + D(x, c_{S^2L}(F(x))))] - \mathbb{E}_{(z,y)\sim p(z,y)}[\min(0, -1 - D(G(z,y), y))]$$

$$\mathcal{L}_G = -\mathbb{E}_{(z,y)\sim p(z,y)}[D(G(z,y), y)],$$

其中 $p(z,y) = p(z)p(y)$，并且 $p(z) = \mathcal{N}(0, I)$ 以及 $p(y)$ 是统一分类。 我们使用缩写S²GAN 作为此方法。

3.2. 协同训练的方法

基于传递的方法的主要缺点是需要通过自我监督训练特征提取器 F 并且学习标签的推理机制（线性分类器或聚类）。在下文中，我们详细介绍了协同训练方法，避免了这两步过程，并学习在 GAN 训练期间推断标签信息。

无监督方法 我们考虑两种方法。在第一个中，我们通过简单地使用相同的标签（见页底 2）对所有真实和生成的示例进行拉伸并从判别器中移除投影层来完全删除标签，即，我们设置 $D(x) = c_{r/f}(\tilde{D}(x))$。对于这种方法，我们使用缩写 SINGLE LABEL 来表示。对于第二种方法，我们将随机标签分配给（未标记的）真实图像。虽然真实图像的标签不能为判别器提供任何有用的信号，但采样标签可能通过提供具有不同于 z 的统计数据的额外随机性来帮助生成器，以及由于嵌入矩阵而产生的额外可训练参数在类条件 BatchNorm 中。此外，虚假数据的标签可以

[2] 请注意，这不一定等同于使用标准（无条件）BatchNorm 替换类条件 BatchNorm，因为本文中使用的条件 BatchNorm 的变体也使用潜在编码的块作为输入; 除了标签信息。

促进判别，因为它们向判别器提供关于假图像的辅助信息。我们将此方法称为 RANDOM LABEL。

半监督方法 当标签可用于实际数据的子集时，我们在 GAN 训练期间直接在判别器的特征表示上训练辅助线性分类器 c_{CT}，并使用它来预测未标记的真实图像的标签。 在这种情况下，判别器损失采取的形式是

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{(x,y)\sim p_{\text{data}}(x,y)}[\min(0, -1 + D(x,y))] \\ &- \lambda\mathbb{E}_{(x,y)\sim p_{\text{data}}(x,y)}[\log p(c_{\text{CT}}(\tilde{D}(x)) = y)] \\ &- \mathbb{E}_{x\sim p_{\text{data}}(x)}[\min(0, -1 + D(x, c_{\text{CT}}(\tilde{D}(x))))] \\ &- \mathbb{E}_{(z,y)\sim p(z,y)}[\min(0, -1 - D(G(z,y), y))], \quad (3)\end{aligned}$$

其中第一项对应于（k%）标记的真实图像的标准条件训练，第二项是标记真实图像上辅助分类器 c_{CT} 的交叉熵损失（权重λ > 0），第三项是无监督的判别器损失，其中未标记的真实图像的标签由 c_{CT} 预测，并且最后一项是生成数据上的标准条件判别器损失。 我们使用缩写S²GAN-CO 作为此方法。 有关说明，请参见图 5。

3.3. GAN 训练期间的自我监督

到目前为止，我们利用自我监督来制定良好的特征表示，或者学习半监督模型（参见第 3.1 节）。 然而，鉴于判别器本身仅仅是分类器，可以通过辅助任务来增加该分类器，即通过旋转预测进行自我监督。 这种方法已经在（Chen 等，2019b）中进行了探讨，其中观察到稳定 GAN 训练。 在这里，我们想评估其影响

当与 3.1 和 3.2 节中介绍的方法结合使用时。 为此，类似于（1）和（2）中的 F 的训练，我们在判别器特征表示$\tilde{D}$上训练附加的线性分类器 c_R，以预测旋转的真实图像x^r的的旋转$r \in \mathcal{R}$和旋转的虚假图像 $G(z,y)^r$。 增加到判别器和生成器损失的相应损失项分别是

$$-\frac{\beta}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log p(c_{\text{R}}(\tilde{D}(x^r)=r)] \quad (4)$$

and

$$-\frac{\alpha}{|\mathcal{R}|}\mathbb{E}_{(z,y)\sim p(z,y)}[\log p(c_{\text{R}}(\tilde{D}(G(z,y)^r)=r)], \quad (5)$$

其中权重 $\alpha,\beta > 0$ 是以平衡损失项。 这种方法如图 6 所示。

4. 实验设置

架构和超参数 GAN 对于训练来说是非常不稳定的，它们的性能在很大程度上取决于神经架构的能力，优化超参数和适当的正则化（Lucic 等，2018; Kurach 等，2018）。我们实现了相应的 BigGAN 架构（Brock et al，2019），它在 ImageNet 上实现了最先进的结果。（见页底 3）我们使用与 Brock 等人相同的优化超参数（2019）。具体来说，我们使用 Adam Optimizer，其生成器的学习率为$5\cdot 10^{-5}$，判别器的学习率为$2\cdot 10^{-4}$（$\beta_1 = 0$，$\beta_2 = 0.999$）。我们在每个生成器步骤之前训练 250k 生成器步骤，并进行 2 次判别器迭代。批量大小固定为 2048，我们使用 120 维的潜码 z。我们在生成器和判别器中使用谱归一化。与 BigGAN 相反，我们不应用正交调节，因为观察到这仅仅略微提高了样本品质（参见 Brock 等人（2019）中的表 1）并且我们不使用截断技巧。

数据集 我们主要关注 IMAGENET，这是通常用于评估 GAN 的最大和最多样化的图像数据集。 IMAGENET 包含 1.3M 训练图像和 50k 测试图像，每个图像对应 1k 对象类别之一。我们将图像大小调整为 128 × 128 × 3，如 Miyato＆Koyama（2018）和 Zhang 等人所做（2018）。通过从每个类中随机选择 k%的样本来获得半监督方法的部分标记的数据集。

[3] 我们剖析了 Brock 等人发布的模型 checkpoint（2019）获得可训练参数及其尺寸的精确计数，并将它们与字节级匹配（参见表 10 和 11）。 我们要强调的是，在这一点上，这种方法是前沿的，成功的最先进的方法不过需要仔细的架构级调整。为了实现可复现性，我们在 Appendix B 中以张量级详细信息详细介绍了这种架构，并在开源时详细介绍了我们的代码：https://github.com/ google/compare_gan.

评估指标 我们使用 Frechet' 起始距离（FID）（Heusel 等，2017）和初始评分（Salimans 等，2016）来评估生成样本的质量。 为了计算 FID，首先将实际数据和生成的样本嵌入预先训练的 Inception 网络的特定层中。然后，多变量高斯拟合数据和距离计算为：

$$\text{FID}(x,g) = ||\mu_x-\mu_g||_2^2 + \text{Tr}(\Sigma_x+\Sigma_g-2(\Sigma_x\Sigma_g)^{\frac{1}{2}}),$$

其中 μ 和 Σ 表示经验均值和协方差，下标 x 和 g 分别表示实际和生成的数据。 FID 被证明对寄生模式（spurious modes）的添加和模式下降（mode dropping）都很敏感（Sajjadi 等，2018; Lucic 等，2018）。初始分数假定含有有意义物体的样品的条件标签分布应具有低熵，样品的可变性应该高，服从以下公式：$\text{IS} = \exp(\mathbb{E}_{x\sim Q}[d_{KL}(p(y\mid x),p(y))])$。虽然它有一些缺陷（Barratt＆Sharma，2018），但我们报告它可以与现有方法进行比较。遵循（Brock 等，2019），使用 50k IMAGENET 测试图像和 50k 随机采样的假图像计算 FID，并且从 50k 随机采样的假图像计算 IS。为 5 个不同的随机抽样假图像集计算所有度量，我们报告了均值。

方法 我们对表 1 中详述的方法进行了广泛的比较，即：未修改的 BIGGAN，无监督方法单标记，随机标记，聚类和半监督方法S^2GAN 和S^2GAN-CO。在所有S^2GAN-CO 实验中，我们使用软标签，即 c_{CT}的 softmax 输出而不是 one-hot 编码的硬估计，正如我们在初步实验中观察到的那样，这稳定了训练。对于S^2GAN，我们默认使用硬标签，但在单独的实验中研究软标签的效果。对于所有半监督方法，我们只能获得 k%的完全真实标签，其中$k \in \{5,10,20\}$。作为附加基线，我们保留 k%标记的真实图像并丢弃所有未标记的真实图像，然后使用剩余的标记图像来训练 BIGGAN（所得模型由 BIGGAN-k％指定）。最后，我们探讨了 GAN 训练期间自我监督对无监督和半监督方法的影响。

我们用不同的随机种子训练每个模型三次，并报告中位数 FID 和中位数 IS。除了 SINGLE LABEL 和 BIGGAN-k% 之外，三次运行的平均值的标准偏差非常低。因此，我们将具有平均 FID 和 IS 值以及标准偏差的表推迟到附录 D 中介绍。所有模型都在 Google TPU v3 Pod 的 128 个核心上进行训练，其中 BatchNorm 统计信息跨核心同步。

无监督方法 对于聚类，我们简单地使用了最好的自监督旋转模型（Kolesnikov 等，2019）。数字

表 1.分析方法的简短摘要。 有关预训练和协同训练方法的详细说明，请分别参见第 3.1 节和第 3.2 节。 第 3.3 节描述了 GAN 训练期间的自我监督。

METHOD	DESCRIPTION
BIGGAN	Conditional (Brock et al., 2019)
SINGLE LABEL	Co-training: Single label
RANDOM LABEL	Co-training: Random labels
CLUSTERING	Pre-trained: Clustering
BIGGAN-$k\%$	Drop all but $k\%$ labeled data
S^2GAN-CO	Co-training: Semi-supervised
S^2GAN	Pre-trained: Semi-supervised
S^3GAN	S^2GAN with self-supervision
S^3GAN-CO	S^2GAN-CO with self-supervision

用于聚类的簇的集合从集合 $\{50, 100, 200, 500, 1000\}$ 中选择。其他无监督的方法没有超参数。

预先训练和协同训练方法 我们采用宽扩展因子 16（Zagoruyko＆Komodakis，2016）的宽 ResNet-50 v2 架构，用于第 3.1 节中描述的预训练方法中的特征提取器 F。我们使用 SGD 在 65 个时期内优化（2）中的损失。批量大小设置为 2048，由 B 未标记的实例和 2048 - B 标记的实例组成。遵循 Goyal 等人的建议（2017）对于大批量训练，我们（i）将学习率设置为 $0.1\frac{B}{256}$，以及（ii）在最初的 5 个时期使用线性学习率预热。学习率在 epoch 45 和 epoch 55 处以 10 倍衰减两次。（2）中的参数 γ 设置为 0.5，每批 B 的未标记示例数为 1536。参数 γ 和 B 调整为从训练集中提取 0.1 % 标记的示例，搜索空间为 $\{0.1, 0.5, 1.0\} \times \{1024, 1536, 1792\}$。这样得到的分类器 $c_{S^2L}(F(x))$ 在 IMAGENET 定值集上的准确度在表 3 中报告。（3）中用于 S^2GAN-CO 损失的参数 γ 从集合 $\{0.1, 0.2, 0.4\}$ 中选择。

GAN 训练期间的自我监督 对于所有方法，我们在（5）中使用（Chen 等，2019b）的推荐参数α = 0.2，并在（4）中进行 β 的小扫描。对于尝试的值（$\{0.25, 0.5, 1.0, 2\}$），我们没有看到巨大的影响，并且对于 S^3GAN 使用β = 0.5。对于 S^3GAN-CO，我们没有重复使用β = 1.0 的扫描。

5. 结果及讨论

回想一下，这项工作的主要目标是在无人监督的情况下匹配（或超出）完全受监督的 BIGGAN，或标记数据的一小部分。在下文中，我们将讨论分析方法在此目标方面的优缺点。

作为基线，我们对 BIGGAN 的重新实现获得了 8.4 的 FID 和 75.0 的 IS，因此就 FID 而言再现了 Brock 等人报道的结果（2019）。我们观察到动态训练的一些差异，我们将在 5.4 节中详细讨论。

5.1 无监督的方法

无监督方法的结果总结在图 7 和表 2 中。完全无监督的 RANDOM LABEL 和 SINGLE LABEL 模型都实现了类似的 FID 为 25 和 IS 为 20。与 BIGGAN 相比，这是一个相当大的差距，并表明额外的监督是必要。我们注意到三个 SINGLE LABEL 模型中的一个崩溃了，而所有三个 RANDOM LABEL 模型都稳定地训练了 250k 次生成器迭代。

使用自我监督对语义表示进行预训练，并通过聚类对该表示的训练数据进行聚类，将 FID 降低约 10%并使 IS 增加约 10%。这些结果是针对 50 个集群获得的，所有其他设置导致更差的结果。虽然这种性能仍然比 BIGGAN 差很多，但这种结果是目前在非预期图像生成方面的最新技术（Chen et al（2019b）报告了无监督生成的 FID 为 33）。

来自聚类的示例图像在补充材料中的图 14,15 和 16 中示出。群集显然是有意义的，并在同一群集中对相似对象进行分组。此外，由聚类有条件地在给定聚类索引上生成的对象反映了属于相应聚类的训练数据的分布。另一方面，我们可以清楚地观察到同一群集中存在多个类。当聚类为 50 个簇时，这是预期的。有趣的是，聚类到更多的聚类（比如 500）会产生类似于 SINGLE LABEL 的结果。

表 2.无监督方法的中位数 FID 和 IS（参见附录中的表 14 中的平均值和标准偏差）。

	FID	IS
RANDOM LABEL	26.5	20.2
SINGLE LABEL	25.3	20.4
SINGLE LABEL (SS)	23.7	22.2
CLUSTERING	23.2	22.7
CLUSTERING (SS)	22.0	23.5

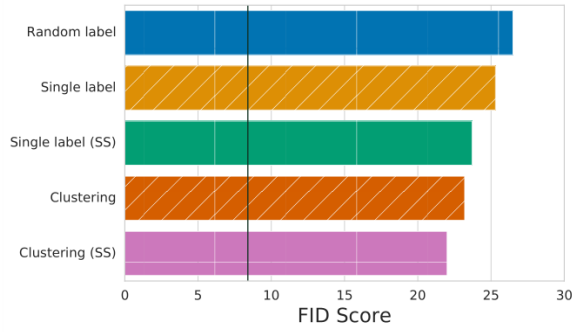


图 7.通过我们的无监督方法获得的中位数 FID。 垂直线表示我们的 BIGGAN 实现的中位数 FID，它使用所有训练图像的标签。虽然无监督和完全监督方法之间的差距仍然很大，但使用预先训练的自我监督表示（聚类）与单标签和随机标签相比可提高样品质量，从而在无监督的情况下实现新的先进技术 在 IMAGENET 上生成。

5.2. 半监督方法

预训练 S^2GAN 模型，我们使用经过自我监督和半监督损失预训练的分类器（参见第 3.1 节），FID 非常小，增加了 10%和 5%真实标签的训练数据，并且当使用 20%的标签时，在 FID 和 IS 方面匹配 BIGGAN（参见表 3）。 我们强调，尽管事实上用于推断标签的分类器对于 5%，10%和 20%标记数据分别具有仅为 50%，63%和 71%的 top-1 精度（参见表 3），与原始标签的 100%相比。 结果显示在表 4 和图 8 中，随机样本和插值可以在补充材料的图 9-17 中找到。

协同训练 我们的协同训练模型S^2GAN-CO 的结果如表 4 所示，它在 GAN 训练期间在判别器表示的基础上训练半监督的线性分类器（参见第 3.2 节）。它可以看出，

表 3.使用 3.1 节中描述的自监督和半监督损失的$c_{S^2L}(F(x))$的 IMAGENET 验证集上的 top-1 和 top-5 错误率（%）。 虽然与完全监督的 IMAGENET 分类任务相比，模型显然不是最先进的，但标签的质量足以匹配并在某些情况下改进最先进的 GAN 自然图像合成。

	Labels		
Metric	5%	10%	20%
Top-1 error	50.08	36.74	29.21
Top-5 error	26.94	16.04	10.33

表 4.预训练与协同训练方法，以及 GAN 训练期间自我监督的效果（参见附录中的表 12 中的平均值和标准差）。 虽然协同训练方法的表现优于完全无监督的方法，但它们明显优于预训练的方法。 GAN 训练期间的自我监督有助于所有情况。

	FID			IS		
	5%	10%	20%	5%	10%	20%
S^2GAN	10.8	8.9	8.4	57.6	73.4	77.4
S^2GAN-CO	21.8	17.7	13.9	30.0	37.2	49.2
S^3GAN	10.4	8.0	7.7	59.6	78.7	83.1
S^3GAN-CO	20.2	16.6	12.7	31.0	38.5	53.1

对于所有考虑的标签百分比，S^2GAN-CO 优于所有完全无监督的方法。虽然S^2GAN-CO 与 5%标签之间的差距和聚类在 FID 方面的差距很小，但S^2GAN-CO 具有相当大的 IS。当使用 20%标记的训练示例时，S^2GAN-CO 获得 13.9 的 FID 和 49.2 的 IS，其与 BIGGAN非常接近并且S^2GAN-CO 给出了S^2GAN 方法的简单性。随着标签百分比的减少，S^2GAN 和S^2GAN-CO 之间的差距也会增大。

有趣的是，即使在 GAN 训练期间被迫学习分类器，S^2GAN-CO 似乎也不像S^2GAN 方法那样稳定训练。随着 BIGGAN-k%的临近，我们只保留标记数据用于训练并丢弃所有未标记的数据，这在 60k 到 120k 迭代后非常不稳定并且崩溃，对于所有三个随机种子和 10%和 20%标记数据都是这样。

5.3 GAN 训练期间的自我监督

到目前为止，我们已经看到预训练的半监督方法，即S^2GAN，能够实现 20%标记数据的最先进性能。在这里，我们调查第 3.3 节中描述的 GAN 训练期间的自我监督是否可以带来进一步的改进。表 4 和图 8 显示了S^3GAN 的实验结果，即S^2GAN 与判别器中的自我监督相结合。

在所有考虑的设置中，自我监督导致 FID 减少和 IS 增加。特别是我们可以将最先进的 BIGGAN 与 10%的标签相匹配，并且在 FID 和 IS 方面均优于使用 20%标签。

对于S^3GAN 而言，由于在 FID 中进行 GAN 训练时的自我监督所带来的改善是相当可观的，大多数情况下约为 10%。调整（4）中判别器自我监督损失的参数没有显著提升

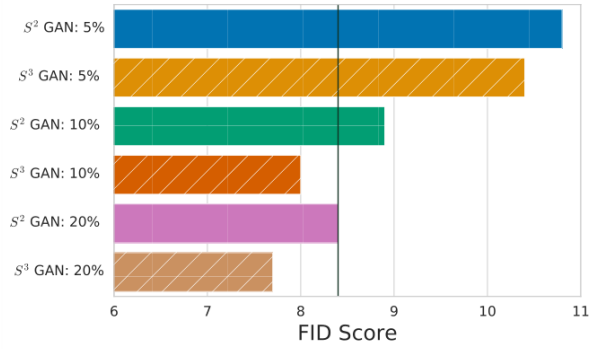


图 8.垂直线表示使用所有标记数据的 BIGGAN 实现的中位数 FID。 建议的S^3GAN 方法能够使用 10%的完全真实标签来匹配最先进的 BIGGAN 模型的性能，并且使用 20%的性能优于它。

在 GAN 训练期间自我监督的好处，至少在所考虑的价值范围内。如表 2 和表 4 所示，GAN 训练期间的自我监督（使用默认参数 α,β）也可使S^2GAN-CO 和 SINGLE LABEL 的改进率提高 5%至 10%。总之，在使用默认参数的 GAN 训练期间进行自我监督可以在所有方法中实现稳定的提升。

5.4 其他见解

软标签的效果 实用者可以选择的设计是S^2GAN 和S^3GAN 是否使用硬标签（即 logits 上的 argmax）或软标签（logits 上的 softmax）（回想一下我们使用软标签默认情况下为S^2GAN-CO 和S^3GAN-CO）。我们最初的期望是，当标签数据非常少时，软标签应该有所帮助，因为软标签带有更多可能被投影判别器利用的信息。令人惊讶的是，表 5 中显示的结果清楚地表明反之亦然。我们目前的假设是，这是由于标签被纳入投影判别器的方式，但我们还没有经验证据。

动态优化 Brock 等（2019）在崩溃之前报告模型的 FID 和 IS，这可以看作是早期停止的一种形式。相比之下，我们设法稳定地训练所提出的模型进行 250k 次生成器迭代。特别是，我们还观察到我们的初代 BIGGAN 实施的稳定训练。 FID 和 IS 作为训练步骤的函数的演变如附录中的图 21 所示。在这一点上，我们只能推测这种差异的起源。

更高的分辨率和低于 5%的标签 以更高的分辨率训练这些模型变得更需要计算，并且需要调整学习速率。我们以 256 × 256 分辨率训练了几个S^3GAN 模型，并在图 12-13 中显示了得到的样本

表 5.使用硬（预测）标签进行训练可获得比使用软（预测）标签进行训练更好的模型（参见附录中表 13 中的平均值和标准偏差）。

	FID			IS		
	5%	10%	20%	5%	10%	20%
S^2GAN	10.8	8.9	8.4	57.6	73.4	77.4
+SOFT	15.4	12.9	10.4	40.3	49.8	62.1

和图 19-20 中的插值。 我们还进行了S^3GAN 实验，其中仅使用 2.5%的标签并且观察到 FID 为 13.6 且 IS 为 46.3。 这表明，给定少量样本，可以明显优于无监督方法（参见图 7）。

6.结论和未来的工作

在这项工作中，我们调查了几种途径，以减少最先进的生成对抗网络中标记数据的需求。我们表明，自我和半监督学习的最新进展可以用于实现新的技术水平，无论是无监督和监督的自然图像合成。

我们相信这是迈向低要求高保真图像合成最终目标的第一步。未来工作有几个重要方向：（i）研究这些技术对更大和更多样化数据集的适用性，以及（ii）研究其他自我和半监督方法对模型质量的影响。 （iii）调查自我监督对其他深度生成模型的影响。最后，我们要强调，与训练大规模生成对抗网络相关的工程挑战可能会阻碍进一步的进展。为帮助缓解此问题并促进可复现性，我们开源了用于实验的所有代码。

致谢

我们要感谢 Ting Chen 和 Neil Houlsby 就自我监督及其对 GAN 的应用进行了富有成效的讨论。我们要感谢 Lucas Beyer，Alexander Kolesnikov 和 Avital Oliver 就自我监督的半监督学习进行了有益的讨论。我们要感谢 Karol Kurach 和 Marcin Michalski 他们对 Compare GAN 库的主要贡献。我们还要感谢 BigGAN 团队（Andy Brock，Jeff Donahue 和 Karen Simonyan）对 TPU 训练 GAN 的见解。最后，我们感谢 Google Brain 团队成员在苏黎世的支持。

参考文献

Agrawal, P., Carreira, J., and Malik, J. Learning to see by moving. In International Conference on Computer Vision, 2015.

Arbel, M., Sutherland, D., Binkowski,´ M. a., and Gretton, A. On gradient regularizers for mmd gans. In Advances in Neural Information Processing Systems. 2018.

Barratt, S. and Sharma, R. A note on the inception score. arXiv preprint arXiv:1801.01973, 2018.

Beyer, L., Kolesnikov, A., Oliver, A., Xiaohua, Z., and Gelly, S. Self-supervised Semi-supervised Learning. In Manuscript in preparation, 2019.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. In International Conference on Learning Representations, 2019.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep 聚类 for unsupervised learning of visual features. European Conference on Computer Vision, 2018.

Chen, T., Lucic, M., Houlsby, N., and Gelly, S. On self modulation for generative adversarial networks. In International Conference on Learning Representations, 2019a.

Chen, T., Zhai, X., Ritter, M., Lucic, M., and Houlsby, N. Self-Supervised GANs via Auxiliary Rotation Loss. In Computer Vision and Pattern Recognition, 2019b.

De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., and Courville, A. C. Modulating early visual pro-cessing by language. In Advances in Neural Information Processing Systems, 2017.

Deng, Z., Zhang, H., Liang, X., Yang, L., Xu, S., Zhu, J., and Xing, E. P. Structured Generative Adversarial Networks. In Advances in Neural Information Processing Systems, 2017.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In International Conference on Computer Vision, 2015.

Dumoulin, V., Shlens, J., and Kudlur, M. A learned repre-sentation for artistic style. In International Conference on Learning Representations, 2017.

Gan, Z., Chen, L., Wang, W., Pu, Y., Zhang, Y., Liu, H., Li, C., and Carin, L. Triangle generative adversarial networks. In Advances in Neural Information Processing Systems, 2017.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In International Conference on Learning Representations, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Advances in Neural Information Processing Systems, 2014.

Goyal, P., Dollar,´ P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch SGD: training imagenet in 1 hour. arXiv preprint arXiv:1706.02677, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a Nash equi-librium. In Advances in Neural Information Processing Systems, 2017.

Jang, E., Devin, C., Vanhoucke, V., and Levine, S. Grasp2Vec: Learning Object Representations from Self-Supervised Grasping. In Conference on Robot Learning, 2018.

Kalchbrenner, N., van den Oord, A., Simonyan, K., Dani-helka, I., Vinyals, O., Graves, A., and Kavukcuoglu, K. Video pixel networks. In International Conference on Machine Learning, 2017.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948, 2018.

Kolesnikov, A., Zhai, X., and Beyer, L. Revisiting Self-supervised Visual Representation Learning. In Computer Vision and Pattern Recognition, 2019.

Kurach, K., Lucic, M., Zhai, X., Michalski, M., and Gelly, S. The GAN Landscape: Losses, architec-tures, regularization, and normalization. arXiv preprint arXiv:1807.04720, 2018.

Lee, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. Unsupervised representation learning by sorting sequences. In International Conference on Computer Vision, 2017.

Li, C., Xu, T., Zhu, J., and Zhang, B. Triple generative adversarial nets. In Advances in Neural Information Pro-cessing Systems. 2017.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bous-quet, O. Are GANs Created Equal? A Large-scale Study. In Advances in Neural Information Processing Systems, 2018.

Menick, J. and Kalchbrenner, N. Generating high fidelity im-ages with subscale pixel networks and multidimensional upscaling. In International Conference on Learning Rep-resentations, 2019.

Miyato, T. and Koyama, M. cgans with projection dis-criminator. In International Conference on Learning Representations, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spec-tral normalization for generative adversarial networks. International Conference on Learning Representations, 2018.

Mundhenk, T. N., Ho, D., and Chen, B. Y. Improvements to context based self-supervised learning. In Computer Vision and Pattern Recognition, 2018.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In European Conference on Computer Vision, 2016.

Odena, A. Semi-supervised learning with generative ad-versarial networks. arXiv preprint arXiv:1606.01583, 2016.

Odena, A., Olah, C., and Shlens, J. Conditional Image Syn-thesis with Auxiliary Classifier GANs. In International Conference on Machine Learning, 2017.

Pinto, L. and Gupta, A. Supersizing self-supervision: Learn-ing to grasp from 50k tries and 700 robot hours. In IEEE International Conference on Robotics and Automation, 2016.

Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. Assessing generative models via precision and recall. In Advances in Neural Information Processing Systems, 2018.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Rad-ford, A., and Chen, X. Improved techniques for training GANs. In Advances in Neural Information Processing Systems, 2016.

Sculley, D. Web-scale k-means 聚类. In International Conference on World Wide Web. ACM, 2010.

Springenberg, J. T. Unsupervised and semi-supervised learn-ing with categorical generative adversarial networks. In International Conference on Learning Representations, 2016.

Sricharan, K., Bala, R., Shreve, M., Ding, H., Saketh, K., and Sun, J. Semi-supervised conditional GANs. arXiv preprint arXiv:1708.05789, 2017.

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. 2016.

Zagoruyko, S. and Komodakis, N. Wide residual networks. British Machine Vision Conference, 2016.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. Self-Attention Generative Adversarial Networks. arXiv preprint arXiv:1805.08318, 2018.

A. 附加样本和插值

图 9.在潜在空间（从左到右）内插时从S^3GAN（20%标签，128 × 128）获得的样本。

图 10.在潜在空间（从左到右）插值时从S^3GAN（20%标签，128 × 128）获得的样本。

图 11.在潜在空间（从左到右）内插时从S^3GAN（20%标签，128 × 128）获得的样本。

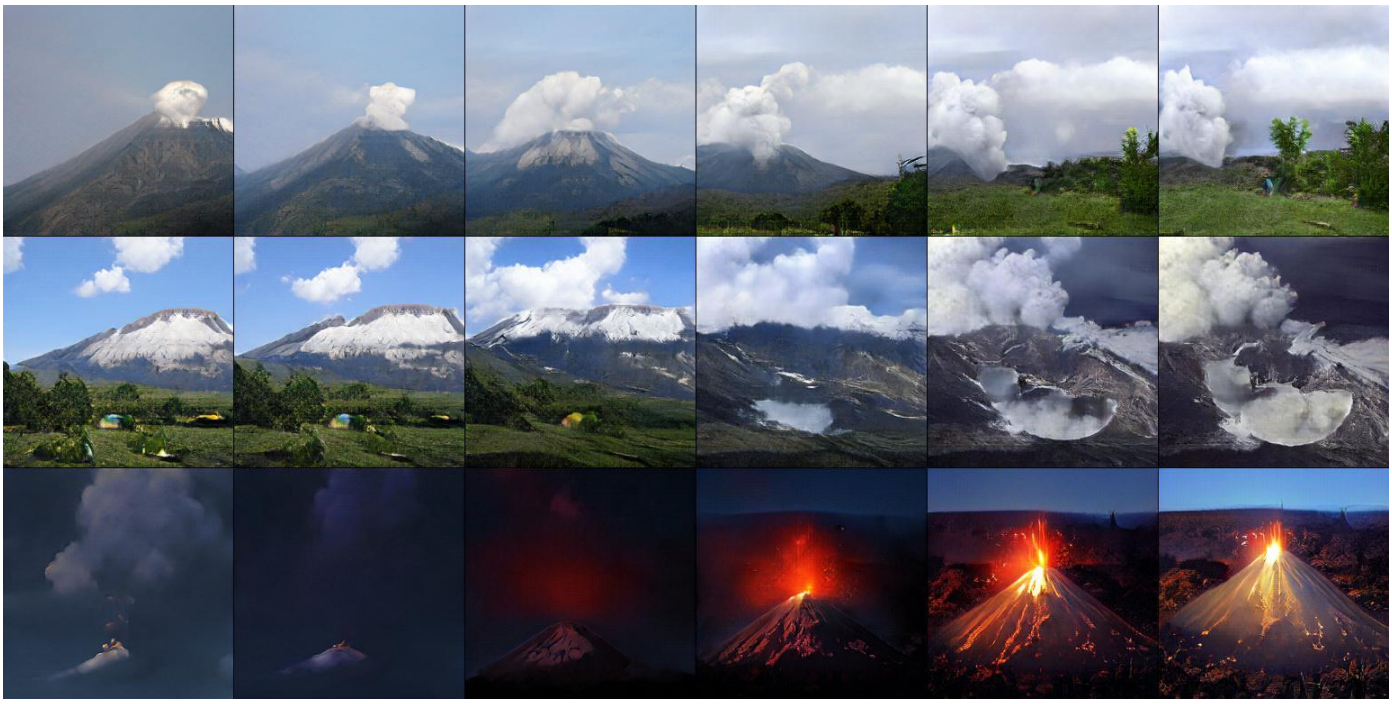


图 12.在潜在空间（从左到右）内插时从S^3GAN（10%标签，256 × 256）获得的样本。

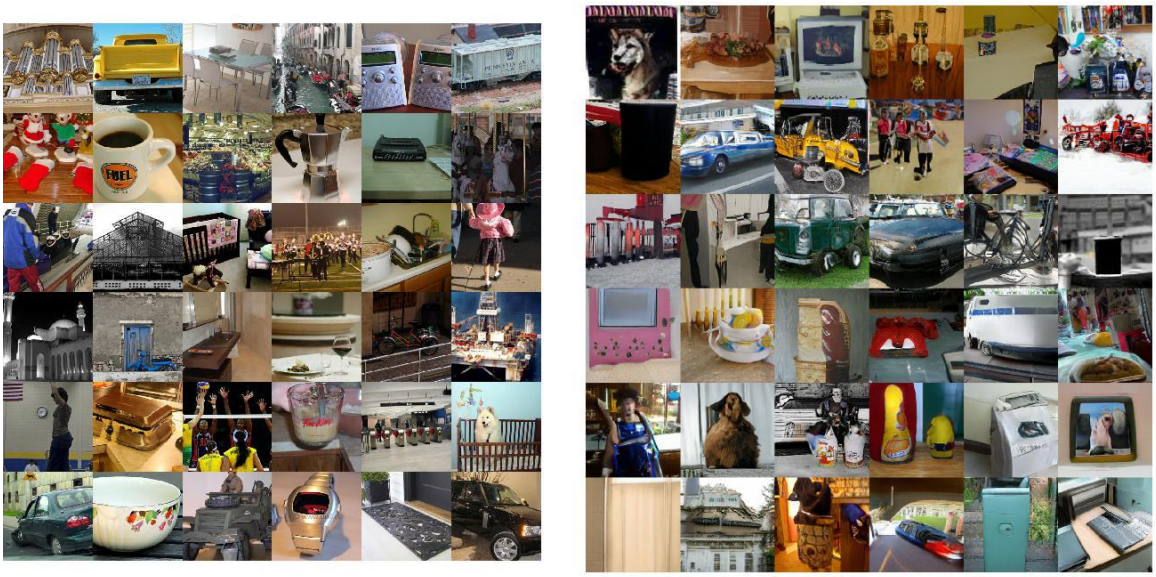
图 13.在潜在空间（从左到右）内插时从S^3GAN（10%标签，256 × 256）获得的样本。

真实图片　　生成图片

图 14.由聚类生成的 50 个簇之一的实际和生成的图像（128 × 128）。 真实和生成的图像都主要显示水下场景。

真实图片 生成图片

图 15.通过聚类生成的 50 个簇之一的实际和生成的图像（128 × 128）。 真实和生成的图像主要显示以不同动物为特色的户外场景。



真实图片 生成图片

图 16.由聚类生成的 50 个簇之一的实际和生成的图像（128 × 128）。 与图 14 和 15 中所示的示例相反，群集显示出不同的室内和室外场景。

用更少的标签生成高保真的图像

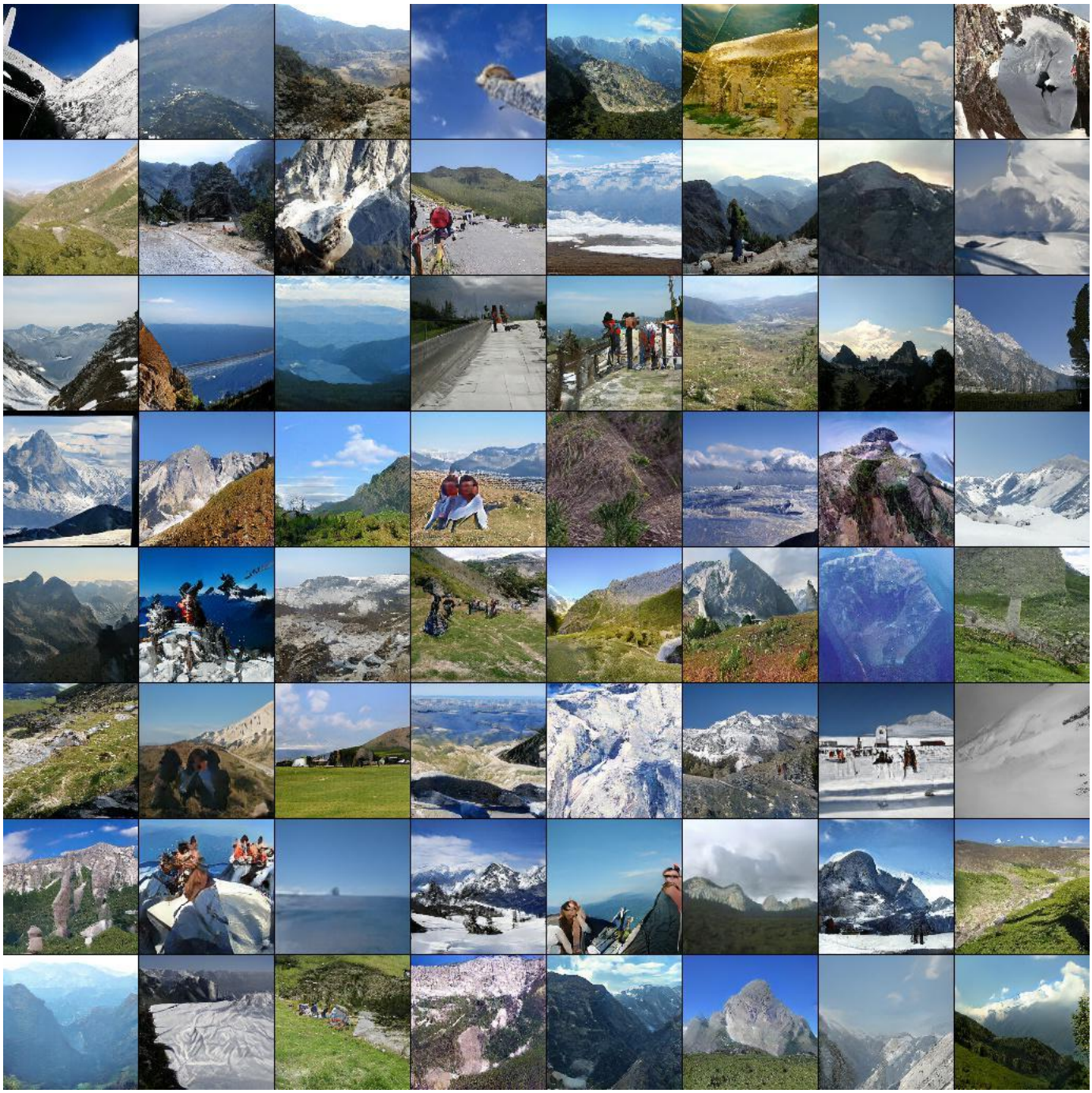


图 17. S^3GAN（20%标签，128 × 128）为单个类生成的样本。 该模型捕捉到了类型上的巨大差异。 人脸和更具动感的场景带来了很多挑战。

用更少的标签生成高保真的图像

图 18.针对不同类别的S³GAN（20%标签，128 × 128）生成的样本。 模型正确地学习了不同的类，我们没有观察到类消失。

用更少的标签生成高保真的图像

图 19.单个类的S^3GAN（10％标签，256 × 256）生成的样本。 该模型捕捉了类别内部的多样性。

图 20. S^3GAN（10%标签，256 × 256）为单个类生成的样本。 该模型捕捉了类别内部的多样性。

B. 架构细节

ResNet 架构是在 Brock 等人之后实施的，表 201 和 7 中描述了（2019）。我们使用缩写 RS 进行重采样，使用 BN 进行批量标准化。 在重新采样列中，我们指示下采样（D）/上采样（U）/无（ - ）设置。 在表 7 中，y 代表标签，h 是来自前面的层的输出（即 pre-logit 层）。 表 8 和表 9 显示了 ResBlock 的详细信息。 加法层通过添加它们来合并快捷路径和卷积路径。 h 和 w 是 ResBlock 的输入高度和宽度，c_i和c_o是 ResBlock 的输入通道和输出通道。 对于没有重新采样的判别器中的最后一个 ResBlock，我们只需从 ResBlock 中删除快捷层。 我们列出了表 10 和表 11 中的所有可训练变量及其形状。

表 6. ResNet 生成器体系结构。 “ch”表示通道宽度乘数，设置为 96。

LAYER	RS	OUTPUT
$z \sim \mathcal{N}(0, 1)$	-	120
Dense	-	$4 \times 4 \times 16 \cdot ch$
ResBlock	U	$8 \times 8 \times 16 \cdot ch$
ResBlock	U	$16 \times 16 \times 8 \cdot ch$
ResBlock	U	$32 \times 32 \times 4 \cdot ch$
ResBlock	U	$64 \times 64 \times 2 \cdot ch$
Non-local block	-	$64 \times 64 \times 2 \cdot ch$
ResBlock	U	$128 \times 128 \times 1 \cdot ch$
BN, ReLU	-	$128 \times 128 \times 3$
Conv $[3, 3, 1]$	-	$128 \times 128 \times 3$
Tanh	-	$128 \times 128 \times 3$

表 7. ResNet 判别器体系结构。 “ch”表示通道宽度乘数，设置为 96。

LAYER	RS	OUTPUT
Input image	-	$128 \times 128 \times 3$
ResBlock	D	$64 \times 64 \times 1 \cdot ch$
Non-local block	-	$64 \times 64 \times 1 \cdot ch$
ResBlock	D	$32 \times 32 \times 2 \cdot ch$
ResBlock	D	$16 \times 16 \times 4 \cdot ch$
ResBlock	D	$8 \times 8 \times 8 \cdot ch$
ResBlock	D	$4 \times 4 \times 16 \cdot ch$
ResBlock	-	$4 \times 4 \times 16 \cdot ch$
ReLU	-	$4 \times 4 \times 16 \cdot ch$
Global sum pooling	-	$1 \times 1 \times 16 \cdot ch$
Sum(embed(y)$\cdot \boldsymbol{h}$)+(dense$\rightarrow$ 1)	-	1

表 8. ResBlock 判别器。

LAYER	KERNEL	RS	OUTPUT
Shortcut	$[1, 1, 1]$	D	$h/2 \times w/2 \times c_o$
BN, ReLU	-	-	$h \times w \times c_i$
Conv	$[3, 3, 1]$	-	$h \times w \times c_o$
BN, ReLU	-	-	$h \times w \times c_o$
Conv	$[3, 3, 1]$	D	$h/2 \times w/2 \times c_o$
Addition	-	-	$h/2 \times w/2 \times c_o$

表 9. ResBlock 生成器。

LAYER	KERNEL	RS	OUTPUT
Shortcut	$[1, 1, 1]$	U	$2h \times 2w \times c_o$
BN, ReLU	-	-	$h \times w \times c_i$
Conv	$[3, 3, 1]$	U	$2h \times 2w \times c_o$
BN, ReLU	-	-	$2h \times 2w \times c_o$
Conv	$[3, 3, 1]$	-	$2h \times 2w \times c_o$
Addition	-	-	$2h \times 2w \times c_o$

NAME	SHAPE	SIZE
discriminator/B1/same conv1/kernel:0	(3, 3, 3, 96)	2,592
discriminator/B1/same conv1/bias:0	(96,)	96
discriminator/B1/down conv2/kernel:0	(3, 3, 96, 96)	82,944
discriminator/B1/down conv2/bias:0	(96,)	96
discriminator/B1/down conv shortcut/kernel:0	(1, 1, 3, 96)	288
discriminator/B1/down conv shortcut/bias:0	(96,)	96
discriminator/non local block/conv2d theta/kernel:0	(1, 1, 96, 12)	1,152
discriminator/non local block/conv2d phi/kernel:0	(1, 1, 96, 12)	1,152
discriminator/non local block/conv2d g/kernel:0	(1, 1, 96, 48)	4,608
discriminator/non local block/sigma:0	()	1
discriminator/non local block/conv2d attn g/kernel:0	(1, 1, 48, 96)	4,608
discriminator/B2/same conv1/kernel:0	(3, 3, 96, 192)	165,888
discriminator/B2/same conv1/bias:0	(192,)	192
discriminator/B2/down conv2/kernel:0	(3, 3, 192, 192)	331,776
discriminator/B2/down conv2/bias:0	(192,)	192
discriminator/B2/down conv shortcut/kernel:0	(1, 1, 96, 192)	18,432
discriminator/B2/down conv shortcut/bias:0	(192,)	192
discriminator/B3/same conv1/kernel:0	(3, 3, 192, 384)	663,552
discriminator/B3/same conv1/bias:0	(384,)	384
discriminator/B3/down conv2/kernel:0	(3, 3, 384, 384)	1,327,104
discriminator/B3/down conv2/bias:0	(384,)	384
discriminator/B3/down conv shortcut/kernel:0	(1, 1, 192, 384)	73,728
discriminator/B3/down conv shortcut/bias:0	(384,)	384
discriminator/B4/same conv1/kernel:0	(3, 3, 384, 768)	2,654,208
discriminator/B4/same conv1/bias:0	(768,)	768
discriminator/B4/down conv2/kernel:0	(3, 3, 768, 768)	5,308,416
discriminator/B4/down conv2/bias:0	(768,)	768
discriminator/B4/down conv shortcut/kernel:0	(1, 1, 384, 768)	294,912
discriminator/B4/down conv shortcut/bias:0	(768,)	768
discriminator/B5/same conv1/kernel:0	(3, 3, 768, 1536)	10,616,832
discriminator/B5/same conv1/bias:0	(1536,)	1,536
discriminator/B5/down conv2/kernel:0	(3, 3, 1536, 1536)	21,233,664
discriminator/B5/down conv2/bias:0	(1536,)	1,536
discriminator/B5/down conv shortcut/kernel:0	(1, 1, 768, 1536)	1,179,648
discriminator/B5/down conv shortcut/bias:0	(1536,)	1,536
discriminator/B6/same conv1/kernel:0	(3, 3, 1536, 1536)	21,233,664
discriminator/B6/same conv1/bias:0	(1536,)	1,536
discriminator/B6/same conv2/kernel:0	(3, 3, 1536, 1536)	21,233,664
discriminator/B6/same conv2/bias:0	(1536,)	1,536
discriminator/final fc/kernel:0	(1536, 1)	1,536
discriminator/final fc/bias:0	(1,)	1
discriminator projection/kernel:0	(1000, 1536)	1,536,000

表 10.包含总共 87,982,370 个参数的判别器的张量描述。

NAME	SHAPE	SIZE
generator/embed_y/kernel:0	(1000, 128)	128,000
generator/fc_noise/kernel:0	(20, 24576)	491,520
generator/fc_noise/bias:0	(24576,)	24,576
generator/B1/bn1/condition/gamma/kernel:0	(148, 1536)	227,328
generator/B1/bn1/condition/beta/kernel:0	(148, 1536)	227,328
generator/B1/up_conv1/kernel:0	(3, 3, 1536, 1536)	21,233,664
generator/B1/up_conv1/bias:0	(1536,)	1,536
generator/B1/bn2/condition/gamma/kernel:0	(148, 1536)	227,328
generator/B1/bn2/condition/beta/kernel:0	(148, 1536)	227,328
generator/B1/same_conv2/kernel:0	(3, 3, 1536, 1536)	21,233,664
generator/B1/same_conv2/bias:0	(1536,)	1,536
generator/B1/up_conv_shortcut/kernel:0	(1, 1, 1536, 1536)	2,359,296
generator/B1/up_conv_shortcut/bias:0	(1536,)	1,536
generator/B2/bn1/condition/gamma/kernel:0	(148, 1536)	227,328
generator/B2/bn1/condition/beta/kernel:0	(148, 1536)	227,328
generator/B2/up_conv1/kernel:0	(3, 3, 1536, 768)	10,616,832
generator/B2/up_conv1/bias:0	(768,)	768
generator/B2/bn2/condition/gamma/kernel:0	(148, 768)	113,664
generator/B2/bn2/condition/beta/kernel:0	(148, 768)	113,664
generator/B2/same_conv2/kernel:0	(3, 3, 768, 768)	5,308,416
generator/B2/same_conv2/bias:0	(768,)	768
generator/B2/up_conv_shortcut/kernel:0	(1, 1, 1536, 768)	1,179,648
generator/B2/up_conv_shortcut/bias:0	(768,)	768
generator/B3/bn1/condition/gamma/kernel:0	(148, 768)	113,664
generator/B3/bn1/condition/beta/kernel:0	(148, 768)	113,664
generator/B3/up_conv1/kernel:0	(3, 3, 768, 384)	2,654,208
generator/B3/up_conv1/bias:0	(384,)	384
generator/B3/bn2/condition/gamma/kernel:0	(148, 384)	56,832
generator/B3/bn2/condition/beta/kernel:0	(148, 384)	56,832
generator/B3/same_conv2/kernel:0	(3, 3, 384, 384)	1,327,104
generator/B3/same_conv2/bias:0	(384,)	384
generator/B3/up_conv_shortcut/kernel:0	(1, 1, 768, 384)	294,912
generator/B3/up_conv_shortcut/bias:0	(384,)	384
generator/B4/bn1/condition/gamma/kernel:0	(148, 384)	56,832
generator/B4/bn1/condition/beta/kernel:0	(148, 384)	56,832
generator/B4/up_conv1/kernel:0	(3, 3, 384, 192)	663,552
generator/B4/up_conv1/bias:0	(192,)	192
generator/B4/bn2/condition/gamma/kernel:0	(148, 192)	28,416
generator/B4/bn2/condition/beta/kernel:0	(148, 192)	28,416
generator/B4/same_conv2/kernel:0	(3, 3, 192, 192)	331,776
generator/B4/same_conv2/bias:0	(192,)	192
generator/B4/up_conv_shortcut/kernel:0	(1, 1, 384, 192)	73,728
generator/B4/up_conv_shortcut/bias:0	(192,)	192
generator/non_local_block/conv2d_theta/kernel:0	(1, 1, 192, 24)	4,608
generator/non_local_block/conv2d_phi/kernel:0	(1, 1, 192, 24)	4,608
generator/non_local_block/conv2d_g/kernel:0	(1, 1, 192, 96)	18,432
generator/non_local_block/sigma:0	()	1
generator/non_local_block/conv2d_attn_g/kernel:0	(1, 1, 96, 192)	18,432
generator/B5/bn1/condition/gamma/kernel:0	(148, 192)	28,416
generator/B5/bn1/condition/beta/kernel:0	(148, 192)	28,416
generator/B5/up_conv1/kernel:0	(3, 3, 192, 96)	165,888
generator/B5/up_conv1/bias:0	(96,)	96
generator/B5/bn2/condition/gamma/kernel:0	(148, 96)	14,208
generator/B5/bn2/condition/beta/kernel:0	(148, 96)	14,208
generator/B5/same_conv2/kernel:0	(3, 3, 96, 96)	82,944
generator/B5/same_conv2/bias:0	(96,)	96
generator/B5/up_conv_shortcut/kernel:0	(1, 1, 192, 96)	18,432
generator/B5/up_conv_shortcut/bias:0	(96,)	96
generator/final_norm/gamma:0	(96,)	96
generator/final_norm/beta:0	(96,)	96
generator/final_conv/kernel:0	(3, 3, 96, 3)	2,592
generator/final_conv/bias:0	(3,)	3

表 11.包含总共 70,433,988 个参数的生成器的张量描述。

C. FID 和 IS 训练曲线

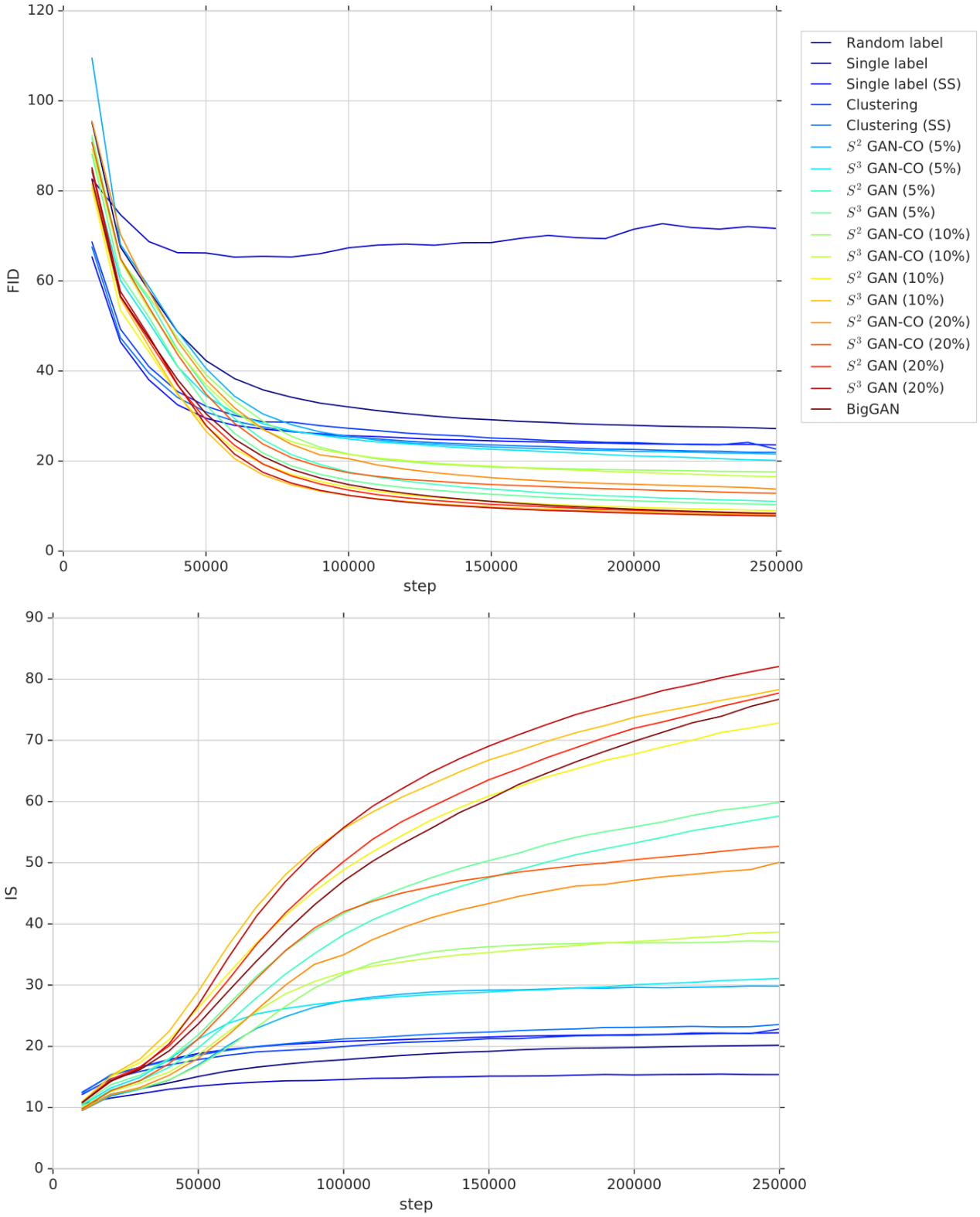


图 21. ImageNet（128 × 128）上的平均 FID 和 IS（3 次运行），用于本文考虑的模型，作为生成器步数的函数。 除了 SINGLE LABEL（一次运行崩溃）之外，所有模型都能稳定地训练。

D. FID 和 IS：平均值和标准偏差

表 12.预训练与协同训练方法，以及在 GAN 训练期间自我监督的效果。 虽然协同训练方法的表现优于完全无监督的方法，但它们明显优于预训练的方法。 GAN 训练期间的自我监督有助于所有情况。

	FID			IS		
	5%	10%	20%	5%	10%	20%
S^2GAN	11.0±0.31	9.0±0.30	8.4±0.02	57.6±0.86	72.9±1.41	77.7±1.24
S^2GAN-CO	21.6±0.64	17.6±0.27	13.8±0.48	29.8±0.21	37.1±0.54	50.1±1.45
S^3GAN	10.3±0.16	8.1±0.14	7.8±0.20	59.9±0.74	78.3±1.08	82.1±1.89
S^3GAN-CO	20.2±0.14	16.5±0.12	12.8±0.51	31.1±0.18	38.7±0.36	52.7±1.08

表 13.使用硬（预测）标签进行训练可获得比使用软（预测）标签进行训练更好的模型。

	FID			IS		
	5%	10%	20%	5%	10%	20%
S^2GAN	11.0±0.31	9.0±0.30	8.4±0.02	57.6±0.86	72.9±1.41	77.7±1.24
S^2GAN SOFT	15.6±0.58	13.3±1.71	11.3±1.42	40.1±0.97	49.3±4.67	58.5±5.84

表 14.无监督方法的平均 FID 和 IS。

	FID	IS
CLUSTERING	22.7±0.80	22.8±0.42
CLUSTERING(SS)	21.9±0.08	23.6±0.19
RANDOM LABEL	27.2±1.46	20.2±0.33
SINGLE LABEL	71.7±66.32	15.4±7.57
SINGLE LABEL(SS)	23.6±0.14	22.2±0.10