

用于高保真自然图像合成的大规模 GAN 训练

Andrew Brock
Heriot-Watt University
ajb5@hw.ac.uk

Jeff Donahue
DeepMind
jeffdonahue@google.com

Karen Simonyan
DeepMind
simonyan@google.com

摘要

尽管最近在生成图像建模方面取得了进展，但是从像 ImageNet 这样的复杂数据集中成功生成高分辨率，多样化的样本仍然是一个难以实现的目标。为此，我们以最大规模培训了生成对抗网络，并研究了这种规模所特有的不稳定性。我们发现将正交正则化应用于生成器使得它适用于简单的“截断技巧”，允许通过截断潜在空间来精确控制样本保真度和变化之间的权衡。我们的修改导致模型在类别条件下的图像合成中达到了新的技术水平。当我们使用 128×128 分辨率在 ImageNet 上进行训练时，我们的模型 (BigGAN) 的 Inception Score (IS) 为 166.3, Fréchet Inception Distance (FID) 为 9.6，相比之前的最佳 IS 为 52.52, FID 为 18.65。

1 介绍



图 1：由我们的模型生成的类别条件下的样本

近年来，生成图像建模的状态发展迅速，生成对抗网络 (GANs, Goodfellow 等人, 2014) 处于使用直接从数据中学习的模型生成高保真、多样化图像的最前沿架构。GANs 训练是动态的，并且几乎对其设置的各个方面都很敏感（从优化参数到模型架构），不过大量的研究已经在经验和理论上都给出了证实，表明 GANs 可以在各种环境中进行稳定的训练。尽管取得了这些进展，但是在条件 ImageNet 下建模 (Zhang 等人, 2018) 的现有实际技术水平只达到了 52.5 的 IS (Salimans 等人, 2016)，而真实数据的 IS 值则为 233。

在这项工作中，我们着手缩小 GAN 生成的图像与 ImageNet 数据集中的真实图像之间的保真度和变化差距。我们为此目标做出以下三个贡献：

- 我们证明了 GAN 从缩放中获益匪浅，并且与现有技术相比，训练模型的参数为 2 到 4 倍，batch 大小达到 8 倍。我们介绍了两种简单的通用体系结构更改，可以提高可伸缩性，并修改正则化方案不断调节，从而显著提升性能。

- 作为我们修改的副作用，我们的模型变得适合“截断技巧”，这是一种简单的采样技术，可以对样本种类和保真度之间的权衡进行明确、细粒度的控制。
- 我们发现特定于大规模 GAN 的不稳定性，并根据经验表征它们。利用此分析的见解，我们证明新颖技术和现有技术的结合可以减少这些不稳定性，但完全的训练稳定性只能以极高的性能成本实现。

我们的修改大大改善了类别条件下的 GAN。当我们在 128×128 分辨率下对 ImageNet 进行训练时，我们的模型（BigGAN）将最先进的 IS 和 FID 分别从 52.52 和 18.65 提高到 166.3 和 9.6。我们还成功地在 ImageNet 上以 256×256 和 512×512 分辨率训练 BigGAN，并且在 256×256 处实现了 TS 和 FID 为 233.0 和 9.3 以及在 512×512 处的 IS 和 FID 为 241.4 和 10.9。最后，我们在更大的数据集上训练我们的模型 - JFT-300M - 并证明我们的设计选择在 ImageNet 传输良好。

2 背景

生成对抗网络 (GAN) 涉及生成器 (G) 和鉴别器 (D) 网络，其目的分别是将随机噪声映射到样本并区分实际和生成的样本。正式地，GAN 目标，其原始形式 (Goodfellow 等人, 2014) 表述为以下两个玩家最小 - 最大问题找到纳什均衡的问题：

$$\min_G \max_D \mathbb{E}_{x \sim q_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))] , \quad (1)$$

其中 $z \in \mathbb{R}^{d_z}$ 是从分布 $P(z)$ 得出的潜在变量，如 $N(0, I)$ 或 $U[-1, 1]$ 。当应用于图像时，G 和 D 通常是卷积神经网络 (Radford 等人, 2016)。如果没有辅助稳定技术，这种训练程序非常脆弱，需要精细调整超参数和架构选择才能工作。

因此，许多最近的研究集中于对 Vanilla GAN 程序的修改以赋予稳定性，借鉴越来越多的经验和理论见解 (Nowozin 等人, 2016; Sønderby 等人, 2017; Fedus 等人, 2018)。一项重点工作是改变目标函数 (Arjovsky 等人, 2017; Mao 等人, 2016; Lim & Ye, 2017; Bellemare 等人, 2017; Salimans 等人, 2018)，以鼓励收敛。另一条专注于通过梯度惩罚来约束 D (Gulrajani 等人, 2017; Kodali 等人, 2017; Mescheder 等人, 2018) 或规范化 (Miyato 等人, 2018)，以抵消无界限的使用损失函数并确保 D 为 G 提供梯度引导。

与我们的工作特别相关的是光谱归一化 (Miyato 等人, 2018)，其通过对参数进行归一化并利用其第一奇异值的运行估计来强制执行 Lipschitz 连续性，从而引入自适应地调整顶部奇异方向的向后动态变化。相关的 (Odena 等人, 2018) 分析 G 的雅可比行列式的条件数，并发现性能取决于 G 的条件。 (Zhang 等人, 2018) 发现在 G 中采用光谱归一化提高了稳定性，允许每次迭代更少的 D 步骤。我们对这些分析进行了扩展，以进一步了解 GAN 训练的理论。

其他工作集中在架构的选择上，例如 SAGAN (Zhang 等人, 2018)，它增加了 (Wang 等人, 2018) 的 self-attention 模块，以提高 G 和 D 模拟全局结构的能力。ProGAN (Karras 等人, 2018) 通过在一系列不断增加的分辨率上训练单个模型来训练单级设置中的高分辨率 GAN。

在有条件的 GAN，即 cGAN 中 (Mirza & Osindero, 2014)，类信息可以以各种方式嵌入模型。在 (Odena 等人, 2017) 中，通过将 1-hot 类向量连接到噪声向量来提供给 G，并且修改目标以鼓励条件样本最大化，并由辅助分类器预测对应类的概率。 (de Vries 等人, 2017) 和 (Dumoulin 等人, 2017) 通过在 BatchNorm (Ioffe & Szegedy, 2015) 层中提供类条件增益和偏差来修改类调节传递给 G 的方式。在 (Miyato & Koyama,

Batch	Ch.	Param (M)	Shared	Hier.	Ortho.	$I_{tr} \times 10^3$	FID	IS
256	64	81.5	SA-GAN Baseline			1000	18.65	52.52
512	64	81.5	X	X	X	1000	15.30	58.77(± 1.18)
1024	64	81.5	X	X	X	1000	14.88	63.03(± 1.42)
2048	64	81.5	X	X	X	732	12.39	76.85(± 3.83)
2048	96	173.5	X	X	X	295(± 18)	9.54(± 0.62)	92.98(± 4.27)
2048	96	160.6	✓	X	X	185(± 11)	9.18(± 0.13)	94.94(± 1.32)
2048	96	158.3	✓	✓	X	152(± 7)	8.73(± 0.45)	98.76(± 2.84)
2048	96	158.3	✓	✓	✓	165(± 13)	8.51(± 0.32)	99.31(± 2.10)
2048	64	71.3	✓	✓	✓	371(± 7)	10.48(± 0.10)	86.90(± 0.61)

表 1：我们提出的修改的模型下的 Fréchet Inception Distance (FID, 越低越好) 和 Inception Score (IS, 越高越好)。Batch 是批量大小, Param 是参数总数, Ch. 是每层中单元数的通道乘数, Shared 表示是否使用共享嵌入, Hier. 是否使用分层潜在空间, Ortho. 是否正交正则化, Itr 如果值为 1000, 则表示该设置对 10^6 次迭代是稳定的, 否则表示在该迭代次数下它就崩溃了。除了行 1-4 之外, 还计算了 8 个不同随机初始化的结果。

2018) 中, 通过使用其特征与一组学习类嵌入之间的余弦相似性作为区分真实和生成样本的附加证据来条件化 D, 有效地鼓励生成其特征与类原型匹配的样本。

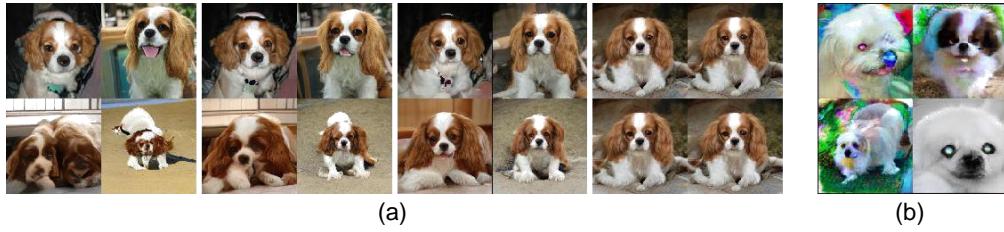
客观评估隐性生成模型很困难 (Theis 等人, 2015)。各种工作已经提出了用于测量模型的样本质量却不具有易处理性的启发式方法 (Salimans 等人, 2016; Heusel 等人, 2017; Binkowski 等人, 2018; Wu 等人, 2017)。其中, Inception Score (IS, Salimans 等人, 2016) 和 Fréchet Inception Distance (FID, Heusel 等人, 2017) 尽管存在明显缺陷 (Barratt & Sharma, 2018), 但已经变得流行。我们将它们用作样本质量的近似度量, 并与以前的工作进行比较。

2 提升 GANS 规模

在本节中, 我们将探索扩大 GAN 训练的方法, 以获得更大型号和更大批量的性能优势。作为基线, 我们采用 (Zhang 等人, 2018) 的 SAGAN 架构。使用铰链损失 (Lim & Ye, 2017; Tran 等人, 2017) 作为 GAN 目标函数。我们使用类条件 BatchNorm (Dumoulin 等人, 2017; de Vries 等人, 2017) 和含投影的 D (Miyato & Koyama, 2018) 向 G 提供类信息。优化设置遵循 (Zhang 等人, 2018) (特别是在 G 中使用 Spectral Norm) 的修改, 我们将学习率减半并且每个 G 步骤采用两个 D 步骤。为了评估, 我们采用了 (Karras 等人, 2018) 的 G 权重移动平均值; (Mescheder 等人, 2018 年) 衰变为 0.9999。我们使用正交初始化 (Saxe 等人, 2014), 而之前的工作使用 $N(0, 0.02I)$ (Radford 等人, 2016) 或 Xavier 初始化 (Glorot & Bengio, 2010)。每个模型都在 Google TPU v3 Pod (Google, 2018) 的 128 到 512 个核心上进行训练, 并在所有设备上计算 G 中的 BatchNorm 统计信息, 而不是像标准实现中那样按设备计算。我们发现, 即使对于我们最大的 512×512 型号, 也不需要逐步增长 (Karras 等人, 2018)。

我们首先增加基线模型的批量大小, 并立即发现这样做的巨大好处。表 1 的第 1-4 行表明, 简单地将批量大小增加 8 倍, 使现有技术 IS 提高了 46%。我们推测这是每批次覆盖更多模式的结果, 为两个网络提供更好的梯度。这种缩放的一个值得注意的副作用是我们的模型在更少的迭代中达到更好的最终性能, 但变得不稳定并且经历完全的训练崩溃。我们将在第 4 节中讨论其原因和后果。对于这些实验, 我们在崩溃后立即停止训练, 并报告之前保存的检查点的分数。

然后, 我们将每层中的宽度 (通道数) 增加 50%, 大约两倍于两个模型中的参数数量。这导致 IS 进一步提高 21%, 我们认为这是由于模型的容量相对于数据集的复杂性而增加。加倍深度似乎不会对 ImageNet 模型产生相同的影响, 反而会降低性能。



(a) 增加截断的影响。从左到右，阈值= 2,1.5,1,0.5,0.04。

(b) 将截断应用于条件差的模型的饱和度假象。

我们注意到用于 G 中的条件 BatchNorm 图层的类嵌入 c 包含大量权重。我们选择使用共享嵌入，而不是为每个嵌入分别设置一个层，这个嵌入会线性投影到每个层的增益和偏差 (Perez 等人, 2018)。这降低了计算和内存成本，并将训练速度（达到给定性能所需的迭代次数）提高了 37%。接下来，我们采用分层潜在空间的变体，其中噪声向量 z 被馈送到 G 的多个层而不仅仅是初始层。这种设计背后的直觉是允许 G 使用潜在空间直接影响不同分辨率和层次结构级别的特征。对于我们的架构，通过将 z 分成每个分辨率的一个块，并将每个块连接到条件向量 c，可以很容易地实现这一点，条件向量 c 被投射到 BatchNorm 的增益和偏差。以前的工作 (Goodfellow 等人, 2014; Denton 等人, 2015) 已经考虑了这个概念的变体；我们的贡献是对此设计的一个小修改。分层延迟可以提高内存和计算成本（主要通过降低第一个线性层的参数预算），提供约 4% 的适度性能提升，并将训练速度提高 18%。

3.1 使用截断技巧处理各种各样的信息

与需要通过其潜在反向传播的模型不同，GAN 可以使用任意先验 $p(z)$ ，但绝大多数先前的工作选择从 $N(0, I)$ 或 $U[-1, 1]$ 。我们质疑这种选择的最优性，并在附录 E 中探索替代方案。

值得注意的是，我们的最佳结果来自于使用与训练中使用的不同的潜在分布进行抽样。采用用 $z \sim N(0, I)$ 训练的模型和从截断的正常值（其中超出范围的值被重新采样以落入该范围内）的样本 z 立即实现对 IS 和 FID 的提升。我们将其称为截断技巧：通过重新调整幅度高于所选阈值的值来截断 z 矢量导致单个样品质量的改善，但代价是整体样品品种的减少。图 2 (a) 证明了这一点：随着阈值的减小，z 的元素被截断为零（潜在分布的模式），各个样本接近 G 的输出分布模式。

这种技术允许对给定 G 的样本质量和变化之间的细粒度，在事后选择进行权衡。值得注意的是，我们可以计算一系列阈值的 FID 和 IS，获得令人联想到精确回忆曲线 (precision-recall curve) 的多样保真曲线 (variety-fidelity curve, 图 16)。由于 IS 不会对类条件模型中缺乏多样性进行惩罚，因此降低截断阈值会导致 IS 的直接增加（类似于精度）。FID 惩罚缺乏多样性（类似于召回）但也奖励精确度，因此我们最初看到 FID 的适度改善，但随着截断接近零并且变化减少，FID 急剧下降。对于许多模型而言，由不同采样引起的分布，相比在训练中看到的会不一样，很容易造成一些麻烦。我们的一些较大模型不适合截断，在馈送截断噪声时会产生饱和伪影(图 2(b))。为了抵消这种情况，我们试图通过将 G 调节为平滑来强制实现截断的适应性，以便 z 的整个空间映射到良好的输出样本。为此，我们转向正交正则化 (Brock 等人, 2017)，它直接强制正交性条件：

$$R_\beta(W) = \beta \|W^\top W - I\|_F^2, \quad (2)$$

其中 W 是权重矩阵和 β 是超参数。众所周知，这种正则化往往过于局限 (Miyato 等人, 2018)，因此我们探索了几种旨在放松约束的变体，同时

仍然为我们的模型赋予了理想的光滑度。我们发现最好的版本从正则化中删除了对角项，并且目标是最小化滤波器之间的成对余弦相似性，但不限制它们的范数：

$$R_\beta(W) = \beta \|W^\top W \odot (\mathbf{1} - I)\|_F^2, \quad (3)$$

其中 $\mathbf{1}$ 表示一个矩阵，其中所有元素都设置为 1。我们扫描 β 值并选择为 10^{-4} ，从而找到足够小的额外正则化，以提高我们的模型易于截断的可能性。在表 1 中，我们观察到没有正交正则化时，只有 16% 的模型适合截断，而有正交正则化训练时则有 60%。

3.2 总结

我们发现当前的 GAN 技术足以扩展到大型模型和分布式大批量训练。我们发现，我们可以显著改进现有技术，并训练模型达到 512×512 分辨率，而无需像 (Karras 等人, 2018) 那样使用明确的多尺度方法。尽管有这些改进，我们的模型依旧经历了训练崩溃，并需要在实践中尽早停止。在接下来的两节中，我们将研究为什么在大规模应用时，以前工作中稳定的设置会变得不稳定。

4 分析

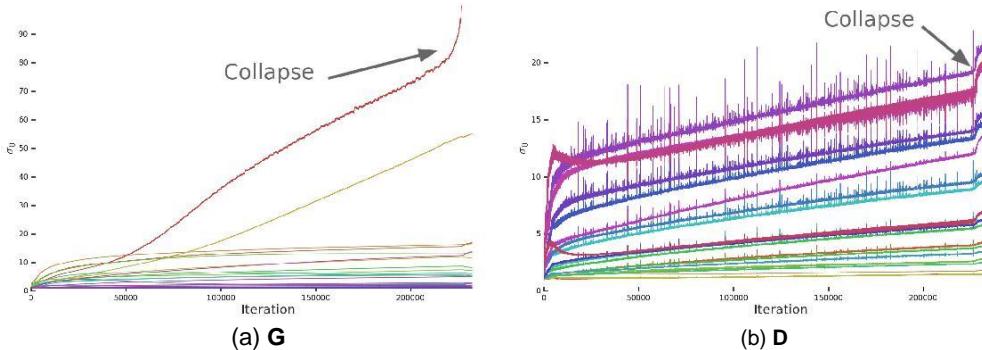


图 3：光谱归一化之前 G (a) 和 D (b) 层中第一个奇异值 σ_0 的典型图。G 中的大多数层都具有良好的光谱，但是没有约束，一个小的子集在整个训练过程中会增长并在崩溃时爆炸。D 的光谱噪声较大，但表现更好。从红色到紫色的颜色表示增加深度。

4.1 表征不稳定性：生成器

以前的许多工作都从各种分析角度和玩家问题上研究了 GAN 的稳定性，但我们观察到的不稳定性发生在小规模稳定的环境中，需要大规模直接分析。我们在训练期间监测一系列权重、梯度和损失统计数据，以寻找可能预示训练崩溃开始的指标，类似于 (Odena 等人, 2018)。我们发现每个权重矩阵中的前三个奇异值 $\sigma_0, \sigma_1, \sigma_2$ 是最有用的。它们可以使用 Arnoldi 迭代方法 (Golub & der Vorst, 2000) 进行有效计算，该方法扩展了 (Miyato 等人, 2018) 使用的功率迭代方法，估计附加的奇异向量和值。如图 3 (a) 和附录 F 所示，出现了清晰的模式：大多数 G 层具有良好的光谱范式，但有些层（通常是在 G 中的第一层，过于完整且非卷积）表现不佳，光谱范式在整个训练过程中增长，在崩溃时爆炸。

为了确定这种症状是否是塌陷造成的或者仅仅是一种症状，我们研究了对 G 施加额外调节以明确抵消光谱爆炸的影响。首先，我们直接

使每个权重的顶部奇异值 σ_0 正则化，朝向固定值 σ_{reg} 或者以某个比率 r 朝向第二奇异值 $r \cdot sg(\sigma_1)$ (其中 sg 为停止梯度操作以防止正则化增加 σ_1)。或者，我们使用部分奇异值分解来代替 σ_0 。给定权重 W ，其第一个奇异向量 u_0 和 v_0 ，以及 σ_0 将被值 σ_{clamp} 钳制，我们的权重变为：

$$W = W - \max(0, \sigma_0 - \sigma_{clamp}) v_0 u_0^\top, \quad (4)$$

其中 σ_{clamp} 被设置为 σ_{reg} 或 $r \cdot sg(\sigma_1)$ 。我们观察到无论有无光谱归一化，这些技术都具有防止 σ_0 或 σ_1 逐渐增加和爆炸的效果，但即使在某些情况下它们可以温和地提高性能，但没有任何组合可以防止训练崩溃。这一证据表明，虽然调节 G 可能会改善稳定性，但它不足以确保稳定性。因此，我们将注意力转向 D。

4.2 表征不稳定性：判别器

与 G 一样，我们分析 D 的权重的光谱以深入了解其行为，然后通过施加额外的约束来寻求稳定训练。图 3 (b) 显示了 D 的 σ_0 的典型图（附录 F 中的附图）。与 G 不同，我们看到光谱是嘈杂的， $\frac{\sigma_0}{\sigma_1}$ 表现良好，并且奇异值在整个训练过程中增长，但只是在崩溃时跳跃而不是爆炸。

D 光谱中的峰值可能表明它周期性地接收到非常大的梯度，但我们观察到 Frobenius 规范是平滑的（附录 F），表明这种效应主要集中在前几个奇异方向上。我们假设这种噪声是通过对抗训练过程进行优化的结果，其中 G 定期产生强烈干扰 D 的 batch。如果这种频谱噪声与不稳定性有因果关系，那么自然的反制是使用梯度惩罚，这明显地规范了 D 的雅可比行列式的变化。我们从 (Mescheder 等人, 2018) 那里探索 R_1 零中心梯度罚分：

$$R_1 := \frac{\gamma}{2} \mathbb{E}_{p_D(x)} [\|\nabla D(x)\|_F^2]. \quad (5)$$

默认建议强度 γ 为 10 时，训练变得稳定并改善 G 和 D 中光谱的平滑度和有界性，但性能严重下降，导致 IS 减少 45%。减少惩罚可以部分缓解这种恶化，但会导致频谱越来越不良；即使将惩罚力度降低到 1（没有发生突然崩溃的最低强度），IS 也会减少 20%。使用正交正则化，DropOut (Srivastava 等人, 2014) 和 L2（详见附录 H）的各种改良重复该实验，揭示了这些正则化策略的行为效果：对 D 的惩罚足够高时，可以实现训练稳定性但是性能成本很高。

我们还观察到 D 在训练期间的损失接近于零，但在崩溃时经历了急剧的向上跳跃（附录 F）。这种行为的一个可能的解释是 D 过度拟合训练集，记忆训练样本而不是学习真实和生成图像之间的一些有意义的边界。作为 D 记忆的简单测试（与 (Gulrajani 等人, 2017) 相关），我们在 ImageNet 训练和验证集上评估未折叠的鉴别器，并测量样本分类为真实或生成的百分比。虽然训练精度始终高于 98%，但验证准确度仅在 50-55% 的范围内，并不比随机猜测更好（无论正则化策略如何）。这证实了 D 确实记住了训练集；我们认为这符合 D 的角色，这不是明确的概括，而是提炼训练数据并为 G 提供有用的学习信号。

4.3 总结

我们发现稳定性不仅仅来自 G 或 D，而是来自他们通过对抗性训练过程的相互作用。虽然他们的不良症状调节可用于追踪和识别不稳定性，但确保合理的调节是训练所必需的，但不足以防止最终的训练崩溃。可以通过强烈约束 D 来强制实现稳定性，但这样做会导致性能上的巨大成本。使用现有技术，可以通过放松这种调节并允许在训练的后期阶段发生塌陷来实现更好的最终性能，此时模型经过充分训练以获得良好的结果。

Model	Res.	FID/IS	(min FID) / IS	FID / (valid IS)	FID / (max IS)
SN-GAN	128	27.62 / 36.80	N/A	N/A	N/A
SA-GAN	128	18.65 / 52.52	N/A	N/A	N/A
BigGAN	128	8.7 ± .6 / 98.8 ± 2.8	7.7 ± .1 / 126.5 ± .1	9.6 ± .4 / 166.3 ± 1	25 ± 2 / 206 ± 2
BigGAN	256	8.2 ± .2 / 154 ± 2.5	7.7 ± .1 / 178 ± 5	9.3 ± .3 / 233 ± 1	25 ± 5 / 295 ± 4
BigGAN	512	10.9 / 154.9	9.3 / 202.5	10.9 / 241.4	24.4 / 275

表 2：不同分辨率下模型的评估。我们报告没有截断的分数（第 3 列），最佳 FID（第 4 列）的分数，验证数据的 IS 分数（第 5 列）和最大 IS 的分数（第 6 列）。在至少三次随机初始化上计算标准偏差。

5 实验



(a) 128×128 (b) 256×256 (c) 512×512 (d)

图 4：来自模型的样本，截断阈值为 0.5 (a-c)，部分训练模型中的类泄漏示例 (d)。

5.1 在 IMAGENET 上的评测

我们使用表 1 第 8 行中的设置，在 Imagenet ILSVRC 2012 (Russakovsky 等人, 2015) 上评估 $128 \times 128, 256 \times 256$ 和 512×512 分辨率的模型。附录 B 中提供了每种分辨率的架构细节。如图 4 所示，附录 A 中有其他样品，我们在表 2 中报告了 IS 和 FID。由于我们的模型能够交换样品品种的质量，因此不清楚如何最好地与现有技术进行比较；因此，我们在三个设置中报告得分值，附录 D 中有详细的曲线。首先，我们报告截断设置下的 FID / IS 值，该值达到最佳 FID。其次，我们在截断设置中报告 FID，我们的模型 IS 与实际验证数据的 IS 相同，推断这是实现最大样本变化的可通过度量，同时仍然达到了良好的“对象性”水平；第三，我们报告每个型号达到的最大 IS 的 FID，以证明必须交换多少品种才能最大限度地提高质量。在所有这三种情况下，我们的模型都优于 (Miyato 等人, 2018 年) 和 (Zhang 等人, 2018) 先前获得的最先进的 IS 和 FID 分数。

我们观察到 D 超过了训练集，加上我们模型的样本质量，提出了一个明显的问题，即 G 是否只记忆训练点。为了测试这一点，我们在像素空间和预训练分类器网络的特征空间中执行分类最近邻分析（附录 A）。此外，我们在图 8 和图 9 中呈现了样本和分类插值（其中 z 保持不变）之间的插值。我们的模型令人信服地在不同的样本之间进行插值，并且其样本的最近邻居在视觉上是不同的，这表明我们的模型不是简单地记住训练数据。

我们注意到，我们部分训练模型的一些失效模式与先前观察到的不同。大多数先前的失败涉及局部人工制品 (Odena 等人, 2016)，由纹理斑点而不是物体组成的图像 (Salimans 等人, 2016)，或规范模式崩溃。我们观察到类泄漏，其中来自一个类的图像包含另一个类的属性，如图 4 (d) 所示。我们还发现，对于我们的模型，ImageNet 上的许多类别比其他类别更难；我们的模型在生成狗（构成数据集的大部分，并且主要通过其纹理区分）方面比人群（其包含数据集的一小部分并且具有更大规模的结构）更成功。附录 A 中提供了进一步的讨论。

Ch.	Param (M)	Shared	Hier.	Ortho.	FID	IS	(min FID) / IS	FID / (max IS)
64	317.1	✗	✗	✗	48.38	23.27	48.6 / 23.1	49.1 / 23.9
64	99.4	✓	✓	✓	23.48	24.78	22.4 / 21.0	60.9 / 35.8
96	207.9	✓	✓	✓	18.84	27.86	17.1 / 23.3	51.6 / 38.1
128	355.7	✓	✓	✓	13.75	30.61	13.0 / 28.0	46.2 / 47.8

表 3: JFT-300M 在 256×256 分辨率下的结果。FID 和 IS 列报告由 JFT-300M 训练的 Inception v2 分类器给出的这些分数，噪声分布为 $z \sim N(0, I)$ (非截断)。 $(\text{min FID}) / \text{IS}$ 和 $\text{FID} / (\text{max IS})$ 列报告最佳 FID 和 IS 的扫描得分，扫描截断噪声分布范围从 $\sigma = 0$ 到 $\sigma = 2$ 。来自 JFT-300M 验证集的图像具有 50.88 的 IS 和 1.94 的 FID。

5.2 在 JFT-300M 上的额外评测

为了确认我们的设计选择对更大、更复杂和对更多样化的数据集有效，我们还在 JFT-300M 的子集上展示了我们系统的结果 (Sun 等人, 2017)。完整的 JFT-300M 数据集包含标有 18K 类别的 300M 真实世界图像。由于类别分布是长尾的，我们对数据集进行子采样以仅保留具有 8.5K 最常见标签的图像。生成的数据集包含 292M 图像 - 比 ImageNet 大两个数量级。对于具有多个标签的图像，我们会在采样图像时随机且独立地采样单个标签。为了计算在该数据集上训练的 GAN 的 IS 和 FID，我们使用在该数据集上训练的 Inception v2 分类器 (Szegedy 等人, 2016)。定量结果如表 3 所示。所有模型均采用 2048 批次进行培训。我们比较了模型的烧蚀版本 - 与 SAGAN (Zhang 等人, 2018) 相当，但批量大小与“完整”相比较使用所有应用技术在 ImageNet 上获得最佳结果的版本（共享嵌入，分层隐藏和正交正则化）。我们的结果表明，即使在相同模型容量 (64 个基本通道) 下设置这么大的数据集，这些技术也能显著提高性能。我们进一步表明，对于这种规模的数据集，我们看到从将模型的容量扩展到 128 个基本通道的显著的额外改进，而对于 ImageNet GAN，额外的容量没有益处。

在图 18 (附录 D) 中，我们给出了在该数据集上训练的模型的截断图。与 ImageNet 不同，截断限制为 0 倾向于产生最高保真度分数，当截断值范围从 0.5 到 1 时，IS 通常最大化我们的 JFT-300M 模型。我们怀疑这至少部分是由于内部 JFT-300M 标签的类可变性，以及图像分布的相对复杂性，其包括具有多种尺度的多个对象的图像。有趣的是，与在 ImageNet 上训练的模型不同，训练倾向于在没有大量正规化的情况下崩溃 (第 4 节)，在 JFT-300M 上训练的模型在数十万次迭代中保持稳定。这表明超越 ImageNet 到更大的数据集可能会部分缓解 GAN 稳定性问题。

我们在此数据集上实现的基线 GAN 模型的改进没有改变基础模型或训练和正则化技术 (超出扩展容量)，证明了我们的研究结果从 ImageNet 扩展到具有规模和复杂性的数据集，迄今为止没有先例图像的生成模型。

6 结论

我们已经证明，对于多个类别的自然图像进行训练而训练的生成对抗网络在保真度和生成样本的多样性方面都非常有利于扩大规模。因此，我们的模型在 ImageNet GAN 模型中创造了新的性能水平，大大提高了现有技术水平。我们还对大规模 GAN 的训练行为进行了分析，并根据其权重的奇异值表征了它们的稳定性，并讨论了稳定性和性能之间的相互作用。.

致谢

我们要感谢 Kai Arulkumaran, Matthias Bauer, Peter Buchlovsky, Jeffrey Defauw, Sander Dieleman, Ian Goodfellow, Ariel Gordon, Karol Gregor, Dominik Grewe, Chris Jones, Jacob Menick, Augustus Odena, Suman Ravuri, Ali Razavi, Mihaela Rosca, 和 Jeff Stanway。

参考文献

- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI'16, pp. 265–283, Berkeley, CA, USA, 2016. USENIX Association. ISBN 978-1-931971-33-1. URL <http://dl.acm.org/citation.cfm?id=3026877>. 3026899.
- Martin Arjovsky, Soumith Chintala, and Leon Bottou. Wasserstein generative adversarial networks. In ICML, 2017.
- Shane Barratt and Rishi Sharma. A note on the Inception Score. In arXiv preprint arXiv:1801.01973, 2018.
- Marc G. Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Remi Munos. The Cramer distance as a solution to biased Wasserstein gradients. In arXiv preprint arXiv:1705.10743, 2017.
- Mikolaj Binkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In ICLR, 2018.
- Andrew Brock, Theodore Lim, J.M. Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. In ICLR, 2017.
- Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In NIPS, 2016.
- Harm de Vries, Florian Strub, Jerémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron Courville. Modulating early visual processing by language. In NIPS, 2017.
- Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In NIPS, 2015.
- Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In ICLR, 2017.
- William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M. Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: GANs do not need to decrease a divergence at every step. In ICLR, 2018.
- Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010.
- Gene Golub and Henk Van der Vorst. Eigenvalue computation in the 20th century. Journal of Computational and Applied Mathematics, 123:35–65, 2000.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, and Aaron Courville. Generative adversarial nets. In NIPS, 2014.
- Google. Cloud TPUs. <https://cloud.google.com/tpu/>, 2018.
- Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of Wasserstein GANs. In NIPS, 2017.

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Gunter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In NIPS, 2017.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.
- Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In ICLR, 2018.
- Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2014.
- Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of GANs. In arXiv preprint arXiv:1705.07215, 2017.
- Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.
- Jae Hyun Lim and Jong Chul Ye. Geometric GAN. In arXiv preprint arXiv:1705.02894, 2017.
- Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Least squares generative adversarial networks. In arXiv preprint arXiv:1611.04076, 2016.
- Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In ICML, 2018.
- Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. In arXiv preprint arXiv:1411.1784, 2014.
- Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In ICLR, 2018.
- Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In ICLR, 2018.
- Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In NIPS, 2016.
- Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. Distill, 2016. doi: 10.23915/distill.00003. URL <http://distill.pub/2016/deconv-checkerboard>.
- Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In ICML, 2017.
- Augustus Odena, Jacob Buckman, Catherine Olsson, Tom B. Brown, Christopher Olah, Colin Raffel, and Ian Goodfellow. Is generator conditioning causally related to GAN performance? In ICML, 2018.
- Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In AAAI, 2018.
- Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. ImageNet large scale visual recognition challenge. IJCV, 115:211–252, 2015.
- Tim Salimans and Diederik Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In NIPS, 2016.
- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In NIPS, 2016.

- Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving GANs using optimal transport. In ICLR, 2018.
- Andrew Saxe, James McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In ICLR, 2014.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
- Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. In ICLR, 2017.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. JMLR, 15:1929–1958, 2014.
- Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In ICCV, pp. 843–852, 2017.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Re-thinking the inception architecture for computer vision. In CVPR, pp. 2818–2826, 2016.
- Lucas Theis, Aaron“ van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In arXiv preprint arXiv:1511.01844, 2015.
- Dustin Tran, Rajesh Ranganath, and David M. Blei. Hierarchical implicit models and likelihood-free variational inference. In NIPS, 2017.
- Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In CVPR, 2018.
- Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger B. Grosse. On the quantitative analysis of decoder-based generative models. In ICLR, 2017.
- Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In arXiv preprint arXiv:1805.08318, 2018.

附录 A 来自 IMAGENET 模型的其他样本，插值和最近邻点



图 5：我们的模型以 256×256 分辨率生成的样本。样品表可在 [here](#) 获得。



图 6：我们的模型以 512×512 分辨率生成的其他样本。

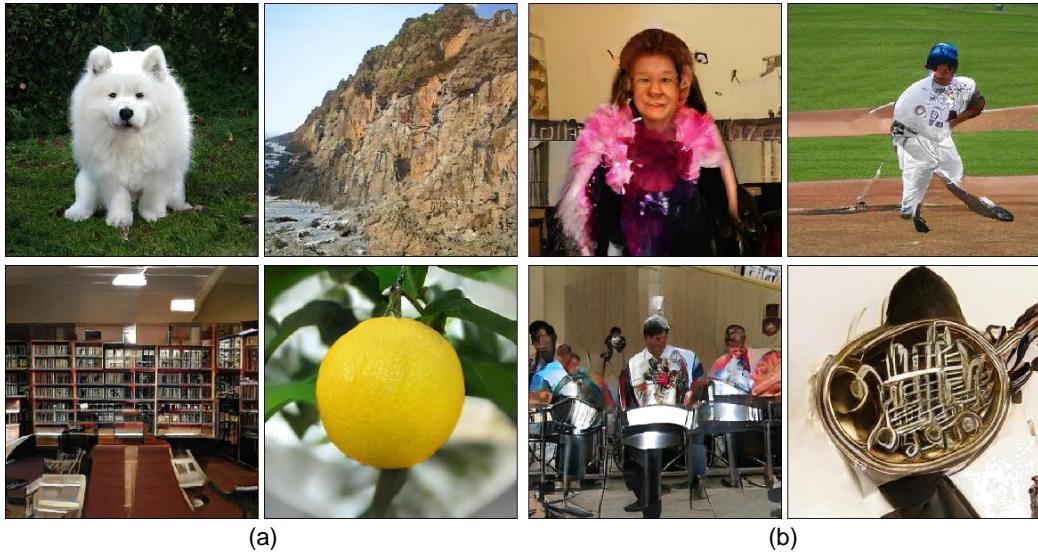


图 7：比较简易类 (a) 和困难类 (b) 在 512×512 分辨率上效果。类似狗的类别很大程度上是纹理的，在数据集中很常见，比涉及未对齐的人脸或人群的类更容易建模。这些类更具动态性和结构性，并且通常具有人类观察者更敏感的细节。在生成高分辨率图像时，即使使用非局部块，也会进一步加剧对全局结构建模的难度。

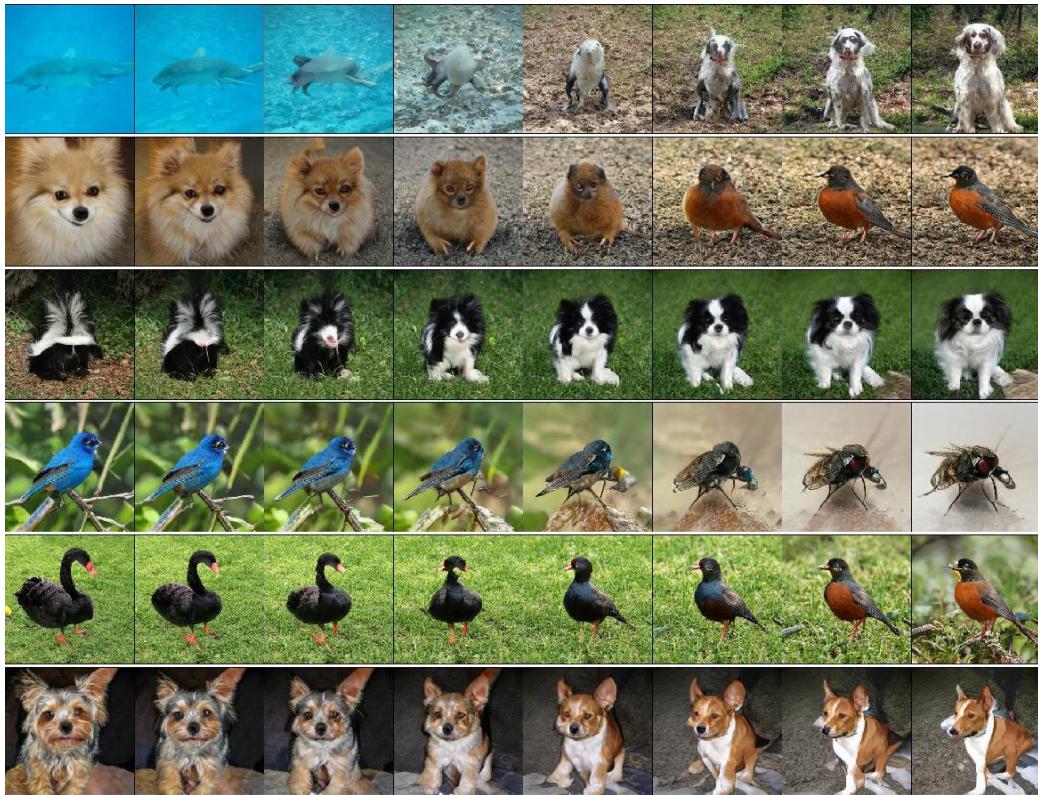


图 8： z, c 对之间的插值。

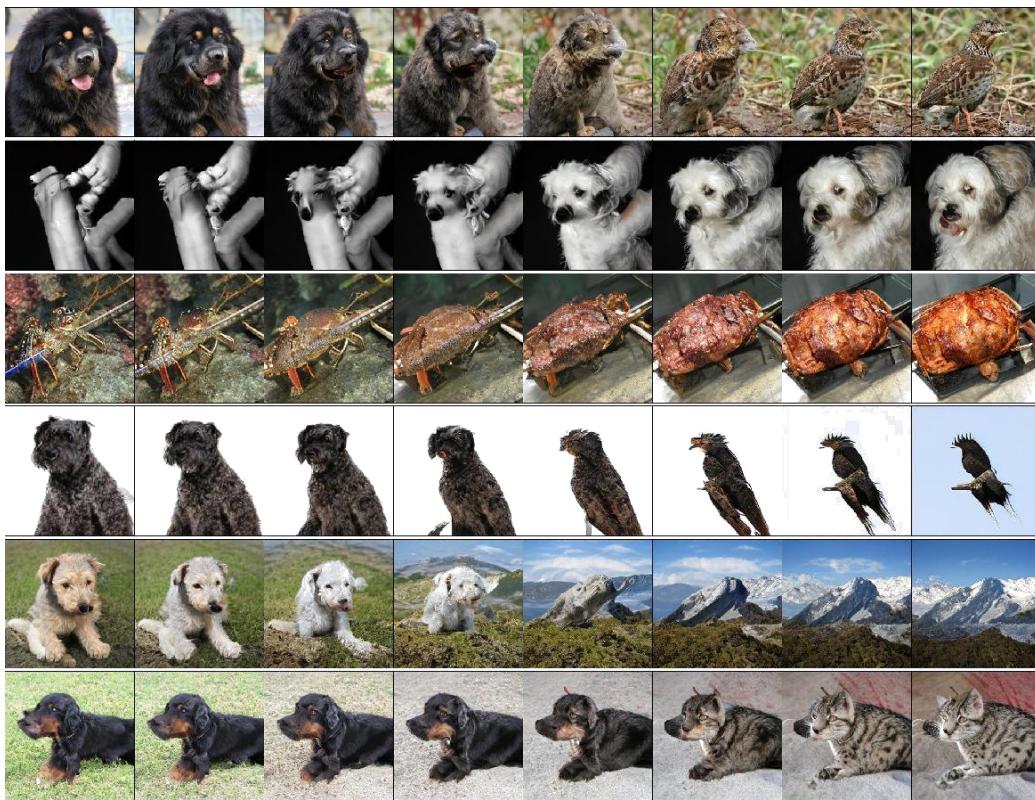


图 9: c 与 z 之间的插值保持不变。端点之间经常保持姿势语义（特别是在最后一行）。第 2 行表明灰度是在关节 z , c 联合空间中编码的，而不是仅在 z 空间。

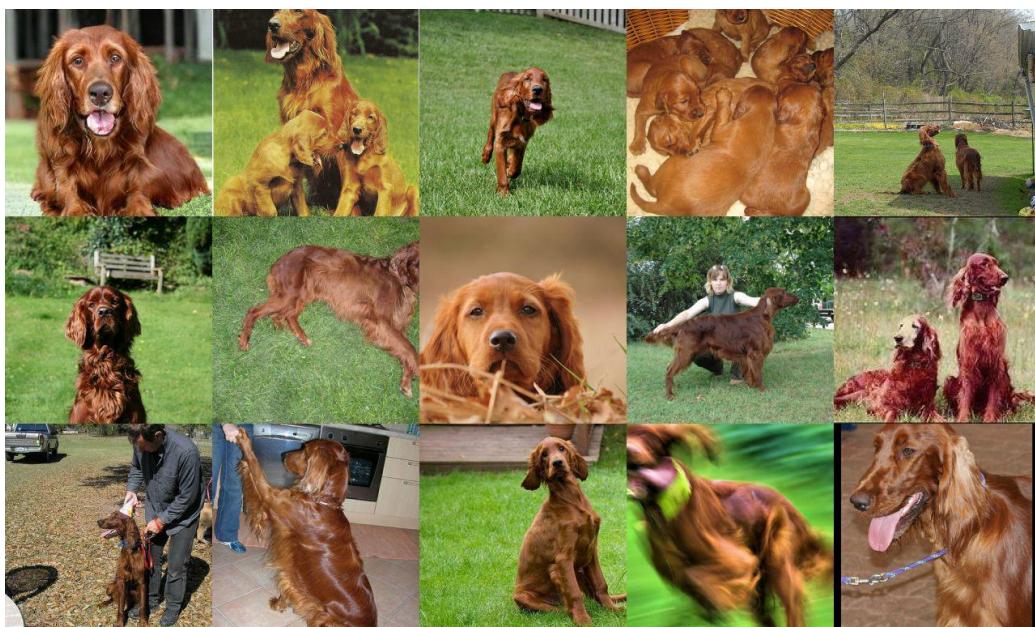


图 10: VGG-16-fc7 中最近邻 (Simonyan & Zisserman, 2015) 的特征空间。生成的图像位于左上角。

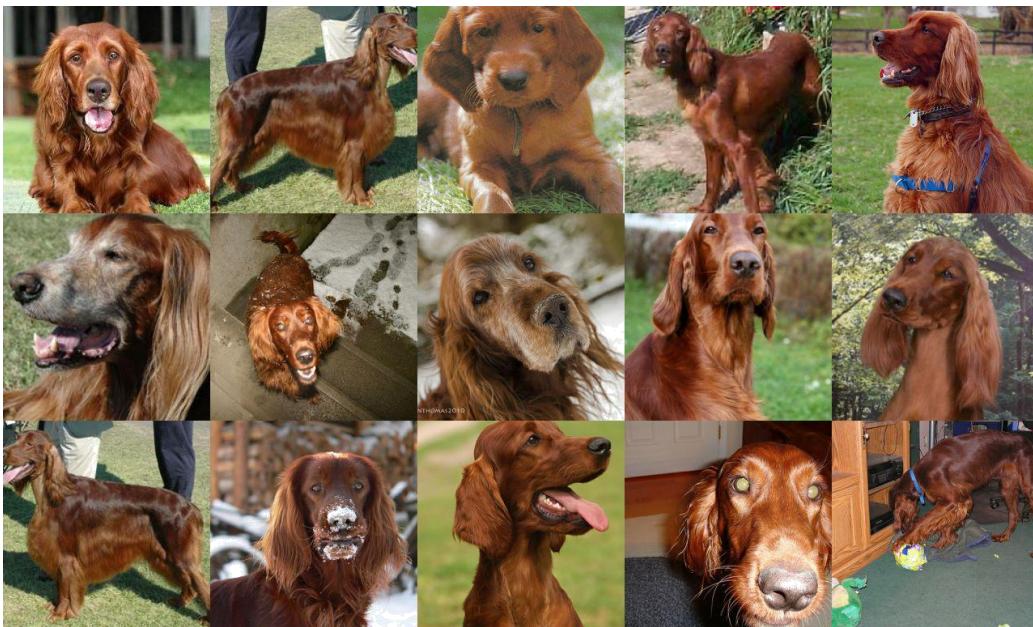


图 11: ResNet-50-avgpool (He 等人, 2016) 中最近邻的特征空间。生成的图像位于左上角。



图 12: 像素空间中最近的邻居。生成的图像位于左上角。



图 13: VGG-16-fc7 (Simonyan&Zisserman, 2015) 中最近邻的特征空间。生成的图像位于左上角。



图 14: ResNet-50-avgpool (He 等人, 2016) 中最近邻的特征空间。生成的图像位于左上角。

附件 B 架构细节

我们使用 ResNet (He 等人, 2016) GAN 架构 (Zhang 等人, 2018)。该架构与 (Miyato 等人, 2018) 使用的架构相同, 但 D 中的信道模式被修改, 使得每个块的第一个卷积层中的滤波器数量等于输出滤波器的数量 (而不是输入过滤器的数量, 如 (Miyato 等人, 2018), (Gulrajani 等人, 2017))。

我们在 G 中使用单个共享类嵌入, 线性投影以生成 BatchNorm 图层的每个样本增益和偏差。偏置投影以零为中心, 而增益投影以一个为中心。当采用分层潜在空间时, 潜在向量 z 沿其信道维度被分成相等大小的块, 并且每个块被单独地连接到传递给给定块的类嵌入的副本。

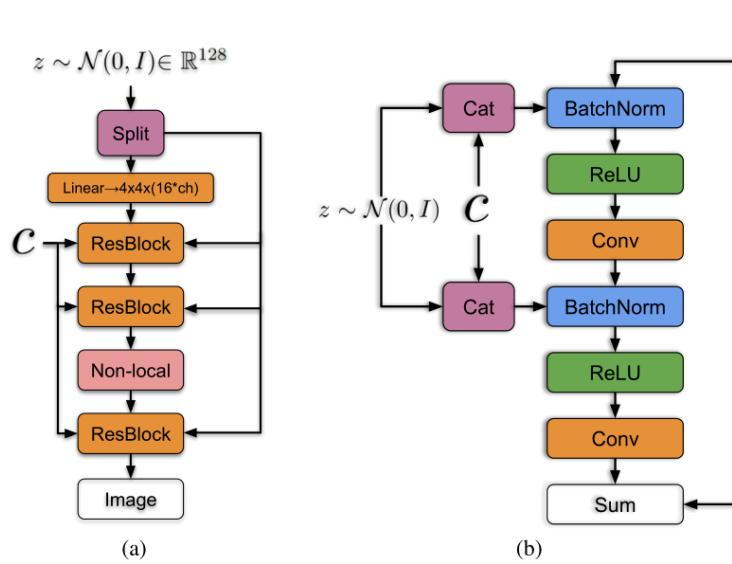


图 15：(a) G 的典型架构布局; 详细信息如下表所示。 (b) G 中的残差块。 c 与 z 块连接并投射到 BatchNorm 的增益和偏差。

表 4: Imagenet 的体系结构, 128×128 像素。“ch” 表示表 1 中每个网络中的通道宽度乘数。

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$	RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
dense, $4 \times 4 \times 16 \cdot ch$	ResBlock down $1 \cdot ch$
ResBlock up $16 \cdot ch$	Non-Local Block (64×64)
ResBlock up $8 \cdot ch$	ResBlock down $2 \cdot ch$
ResBlock up $4 \cdot ch$	ResBlock down $4 \cdot ch$
ResBlock up $2 \cdot ch$	ResBlock down $8 \cdot ch$
Non-Local Block (64×64)	ResBlock down $16 \cdot ch$
ResBlock up $1 \cdot ch$	ResBlock $16 \cdot ch$
BN, ReLU, 3×3 conv 3	ReLU
Tanh	Global sum pooling
(a) Generator	Embed(y)- h + (dense $\rightarrow 1$)
	(b) Discriminator

表 5: ImageNet 的体系结构, 256×256 像素。 “ch” 表示表 1 中每个网络中的通道宽度乘数。相对于 128×128 架构, 我们在 16×16 分辨率的每个网络中添加额外的 8-ch 的 ResBlock, 并将 G 中的非本地块移动到一个阶段以 128×128 分辨率。内存约束阻止我们移动 D 中的非本地块。

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
dense, $4 \times 4 \times 16 \cdot ch$
ResBlock up $16 \cdot ch$
ResBlock up $8 \cdot ch$
ResBlock up $8 \cdot ch$
ResBlock up $4 \cdot ch$
ResBlock up $2 \cdot ch$
Non-Local Block (128×128)
ResBlock up $1 \cdot ch$
BN, ReLU, 3×3 conv 3
Tanh

(a) Generator

RGB image $x \in \mathbb{R}^{256 \times 256 \times 3}$
ResBlock down $1 \cdot ch$
ResBlock down $2 \cdot ch$
Non-Local Block (64×64)
ResBlock down $4 \cdot ch$
ResBlock down $8 \cdot ch$
ResBlock down $8 \cdot ch$
ResBlock down $16 \cdot ch$
ResBlock $16 \cdot ch$
ReLU
Global sum pooling
Embed(y)· \mathbf{h} + (dense $\rightarrow 1$)

(b) Discriminator

表 6: ImageNet 的体系结构, 512×512 像素。 “ch” 表示表 1 中每个网络中的通道宽度乘数。相对于 256×256 架构, 我们在 512×512 阶段添加额外的 1-ch 的 ResBlock。内存限制迫使我们将两个网络中的非本地块移回 64×64 阶段, 如 128×128 像素设置。

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
dense, $4 \times 4 \times 16 \cdot ch$
ResBlock up $16 \cdot ch$
ResBlock up $8 \cdot ch$
ResBlock up $8 \cdot ch$
ResBlock up $4 \cdot ch$
Non-Local Block (64×64)
ResBlock up $2 \cdot ch$
ResBlock up $1 \cdot ch$
ResBlock up $1 \cdot ch$
BN, ReLU, 3×3 conv 3
Tanh

(a) Generator

RGB image $x \in \mathbb{R}^{512 \times 512 \times 3}$
ResBlock down $1 \cdot ch$
ResBlock down $1 \cdot ch$
ResBlock down $2 \cdot ch$
Non-Local Block (64×64)
ResBlock down $4 \cdot ch$
ResBlock down $8 \cdot ch$
ResBlock down $8 \cdot ch$
ResBlock down $16 \cdot ch$
ResBlock $16 \cdot ch$
ReLU
Global sum pooling
Embed(y)· \mathbf{h} + (dense $\rightarrow 1$)

(b) Discriminator

附件 C 实验细节

我们的基本设置遵循 SA-GAN (Zhang 等人, 2018) , 并在 TensorFlow 中实施 (Abadi 等人, 2016) 。我们采用附录 B 中详述的体系结构, 在每个网络的单个阶段插入非本地块。 G 和 D 网络都用正交初始化初始化 (Saxe 等人, 2014) 。我们使用 Adam 优化器 (Kingma & Ba, 2014) , D 的学习率为 $2 \cdot 10^{-4}$, G 的学习率为 $5 \cdot 10^{-5}$;在两个网络中, $\beta_1 = 0$ 和 $\beta_2 = 0.999$ 。我们试验了每 G 步骤的 D 步数 (从 1 到 6 变化), 发现每 G 步的两个 D 步给出了最好的结果。

我们在采样时使用 G 的权重的指数移动平均值, 衰减率设置为 0.9999。我们在 G 中使用交叉副本 BatchNorm (Ioffe & Szegedy, 2015) , 其中批量统计信息在所有设备上聚合, 而不是像标准实现中那样在单个设备上聚合。在 SA-GAN 之后, 在 G 和 D 中使用光谱归一化 (Miyato 等人, 2018) (Zhang 等人, 2018) 。我们使用 Google TPU v3 Pod 进行训练, 其核心数与分辨率成正比: 128×128 为 128, 256×256 为 256, 512×512 为 512。大多型号的训练需要 24 到 48 小时。我们在 BatchNorm 和 Spectral Norm 中将 ϵ 从默认值 10^{-8} 增加到 10^{-4} , 以便缓和低精度数值问题。

我们通过沿长边裁剪来预处理数据, 并通过区域重采样重新缩放到给定的分辨率。由于 ImageNet 数据集具有许多低分辨率图像, 因此直接在 512×512 处进行训练会产生混叠结果, 因此我们会过滤掉所有短边长度小于 400 像素的图像。类似于 (Karras 等人, 2018) 使用的 CelebA-HQ 数据集。这将数据集大小减少到大约 200,000 个实例。

C.1 BATCHNORM 统计和抽样

批量标准化分类器网络的默认行为是在测试时使用激活时刻的运行平均值。以前的工作 (Radford 等人, 2016) 在采样图像时使用批量统计。虽然从技术上讲这不是一种无效的采样方式, 但这意味着结果取决于测试批量大小 (以及分割的设备数量), 并且进一步使再现性变得复杂。

我们发现这个细节非常重要, 测试批量大小的变化会导致性能的急剧变化。当使用 G 的权重的指数移动平均值进行采样时, 这进一步加剧, 因为 BatchNorm 运行平均值是使用非平均权重计算的, 并且是对平均权重的激活统计数据的差估计。

为了解决这两个问题, 我们采用 “常设统计” , 我们通过运行 G 通过多个前向传递 (通常为 100) 来计算采样时间的激活统计数据, 每个传递具有不同批次的随机噪声, 并且存储均值和方差。所有前传传球。类似于使用运行统计数据, 这导致 G 的输出变得对批量大小和设备数量不变, 即使在生成单个样本时也是如此。

C.2 CIFAR-10

我们使用表 1 第 8 行中的设置在 CIFAR-10 (Krizhevsky & Hinton, 2009) 上运行我们的网络, 并实现 IS 为 9.22 和 FID 为 14.73 而不截断。

C.3 IMAGENET 图像的 Inception Score

我们为 ImageNet 的训练和验证集计算 IS。在 128×128 处, 训练数据的 IS 为 233, 并且验证数据的 IS 为 166。在 256×256 处, 训练数据的 IS 为 377, 并且验证数据的 IS 为 234。在 512×512 处, 训练数据 IS 为 348, 验证数据的 IS 为 241。训练和验证分数之间的差异是由于初始分类器已经对训练数据进行了训练, 从而产生了初始分数首选的高置信度输出。

附件 D

附加绘图

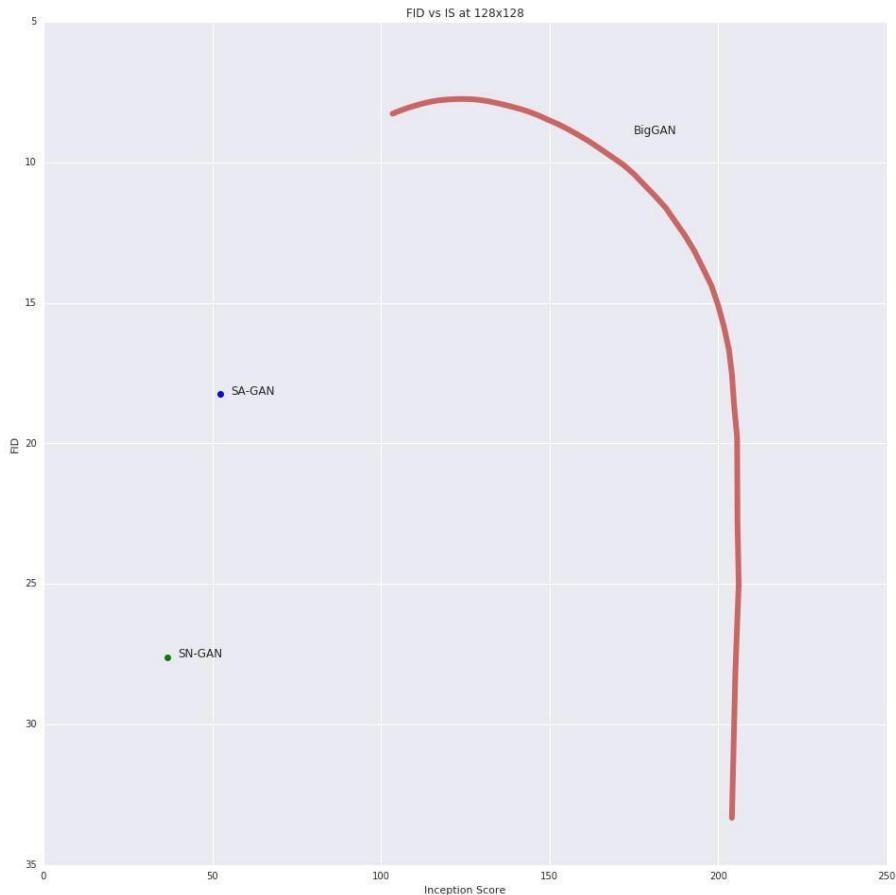


图 16：IS 与 FID 在 128×128 处。分数在三个随机种子中取平均值。

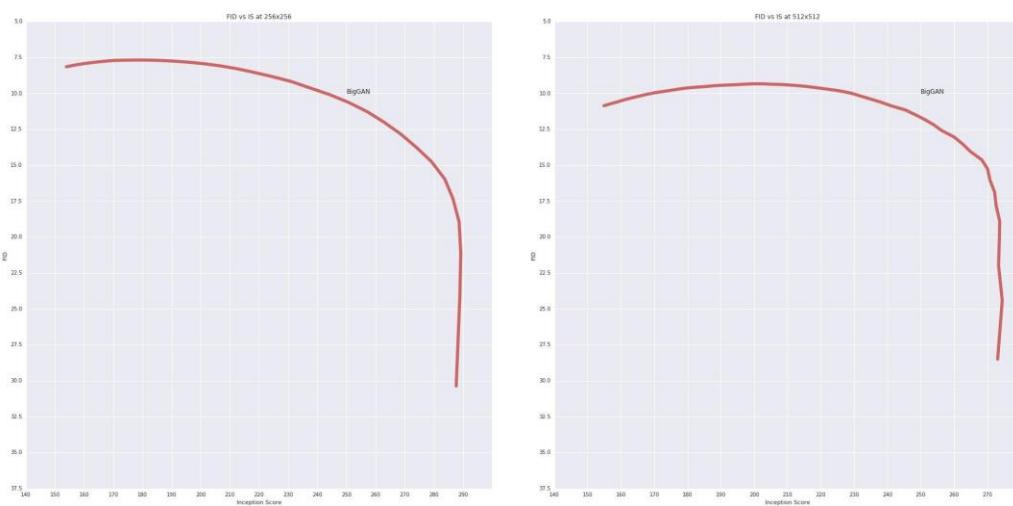


图 17：256 和 512 像素的 IS 与 FID。得分在三个随机种子中平均为 256。

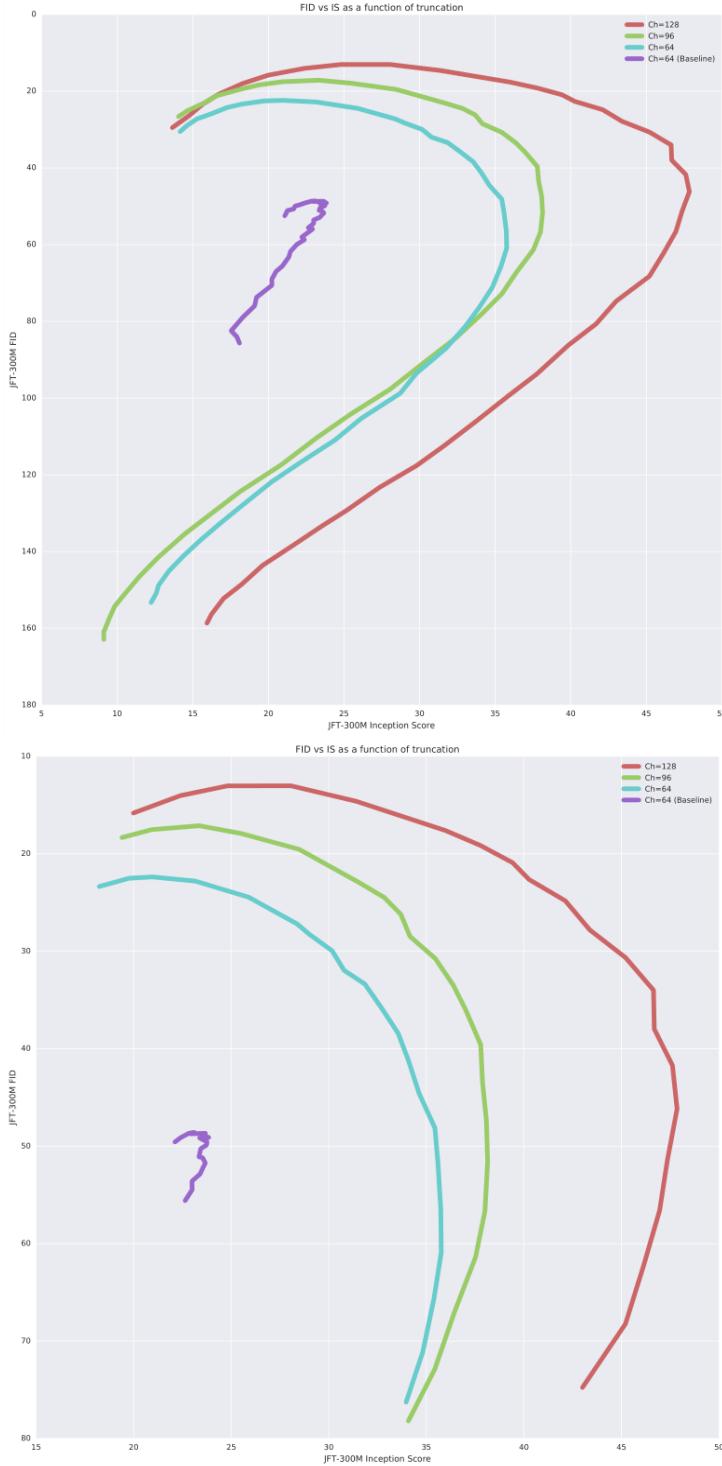


图 18：JFT-300M IS 与 256×256 处的 FID。我们显示截断值从 $\sigma = 0$ 到 $\sigma = 2$ (顶部) 和从 $\sigma = 0.5$ 到 $\sigma = 1.5$ (底部)。每条曲线对应于表 3 中的一行。用基线标记的曲线对应于第一行 (禁用正交正则化和其他技术)，而其余对应于行 2-4 —— 不同容量 (Ch) 的相同架构。

附件 E 选择潜在空间

虽然大多数以前的工作采用 $N(0, I)$ 或 $U[-1, 1]$ 作为 z 的先验（输入到 G 的噪声），我们可以自由选择我们可以采样的任何潜在分布。我们通过考虑一系列可能的设计来探索潜伏的选择，如下所述。对于每个潜在的，我们提供其设计背后的直觉，并简要描述它在 SAGAN 基线中用 $z \sim N(0, I)$ 直接替代的表现。由于截断技巧被证明比切换到这些潜伏中的任何一个更有益，我们不进行完全消融研究，并且使用 $z \sim N(0, I)$ 来获得我们的主要结果以充分利用截断。我们发现在没有截断的情况下效果最好的两个潜伏者是 Bernoulli{0, 1} 和 Censored Normal $\max(N(0, I), 0)$ ，两者都提高了训练速度并轻微提高了最终性能，但不太容易截断。

我们还消除了潜在空间维度的选择（默认情况下是 $z \in \mathbb{R}^{128}$ ），发现我们能够成功地训练潜在维度低至 $z \in \mathbb{R}^8$ ，并且对于 $z \in \mathbb{R}^{32}$ 我们看到性能的最小下降。虽然这比以前的许多作品要小得多，但直接比较单级网络（例如（Karras 等人，2018），在高度约束的数据集中使用带有 30,000 张图像的 $z \in \mathbb{R}^{512}$ 潜在空间）是不合适的，我们的网络提供了额外的分类信息作为输入。

潜在空间

- **$N(0, I)$** 。我们在主要实验中使用的潜在空间的标准选择。
- **$U [-1, 1]$** 。另一个标准选择；我们发现它的表现与 $N(0; I)$ 类似。
- **Bernoulli {0, 1}**。离散的潜在可能反映了我们之前的自然图像变化的潜在因素不是连续的，而是离散的（一个特征存在，另一个特征不存在）。这个潜伏优于 $N(0, I)$ （以 IS 表示）8%，并且迭代次数减少 60%。
- **max ($N(0, I)$, 0)**，也称为截尾正态。这个潜伏设计用于在潜在空间中引入稀疏性（反映我们之前的某些潜在特征有时存在，有时不存在），但也允许那些潜伏物连续变化，表现出活跃的潜伏期的不同程度的强度。这个潜伏优于 $N(0, I)$ （就 IS 而言）15-20%，并且往往需要更少的迭代。
- **Bernoulli {-1, 1}**。这个潜在的设计是离散的，但不是稀疏的（因为网络可以学习激活以响应负输入）。该潜伏性与 $N(0, I)$ 几乎相同。
- **{-1, 0, 1} 中等概率的独立分类**。选择该分布是离散的并且具有稀疏性，但也允许潜伏者采用正值和负值。该潜伏性与 $N(0, I)$ 几乎相同。
- **$N(0, I)$ 乘以 Bernoulli {0, 1}**。选择该分布以具有连续的潜在因子，其也是稀疏的（峰值为零），类似于截尾正常但不限于为正。该潜伏性与 $N(0, I)$ 几乎相同。
- **连接 $N(0, I)$ 和 Bernoulli {0, 1}**，每个占潜在尺寸的一半。这是受（Chen 等人，2016）的启发，并选择允许一些变异因素是离散的，而其他因素是连续的。这个潜伏优于 $N(0, I)$ 约 5%。
- **方差退火**：我们从 $N(0, \sigma I)$ 采样，允许 σ 在训练中变化。我们比较了各种分段时间表，发现在训练过程中从 $\sigma = 2$ 开始并退回到 $\sigma = 1$ ，性能略有提高。可能的差异计划空间很大，我们没有深入探讨-我们怀疑更有原则或更好调整的计划可能会更强烈地影响绩效。
- **每样本变量方差**： $\mathcal{N}(0, \sigma_i I)$ ，其中 $\sigma_i \sim \mathcal{U}[\sigma_l, \sigma_h]$ 对于批次中的每个样品 i 独立地，并且 (σ_l, σ_h) 是超参数。选择该分布以通过以非恒定方差馈送网络噪声样本来尝试并改善对截断技巧的适应性。这似乎并没有影响性能，但我们没有深入探讨它。人们也可以考虑调度 (σ_l, σ_h) ，类似于方差退火。

附录 F 训练数据监测

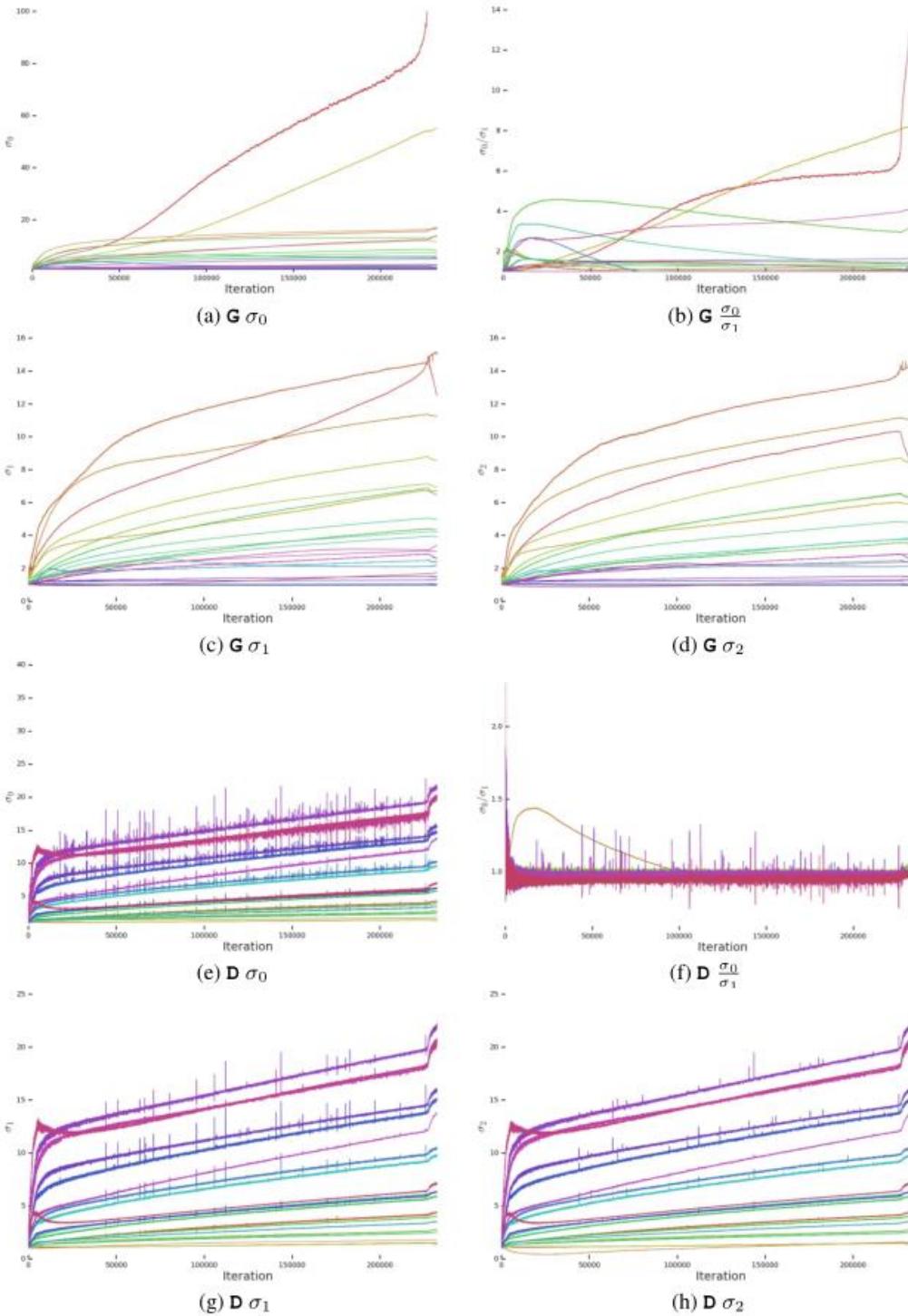


图 19：没有特殊修改的典型模型的培训统计数据。在 200000 次迭代后发生折叠。

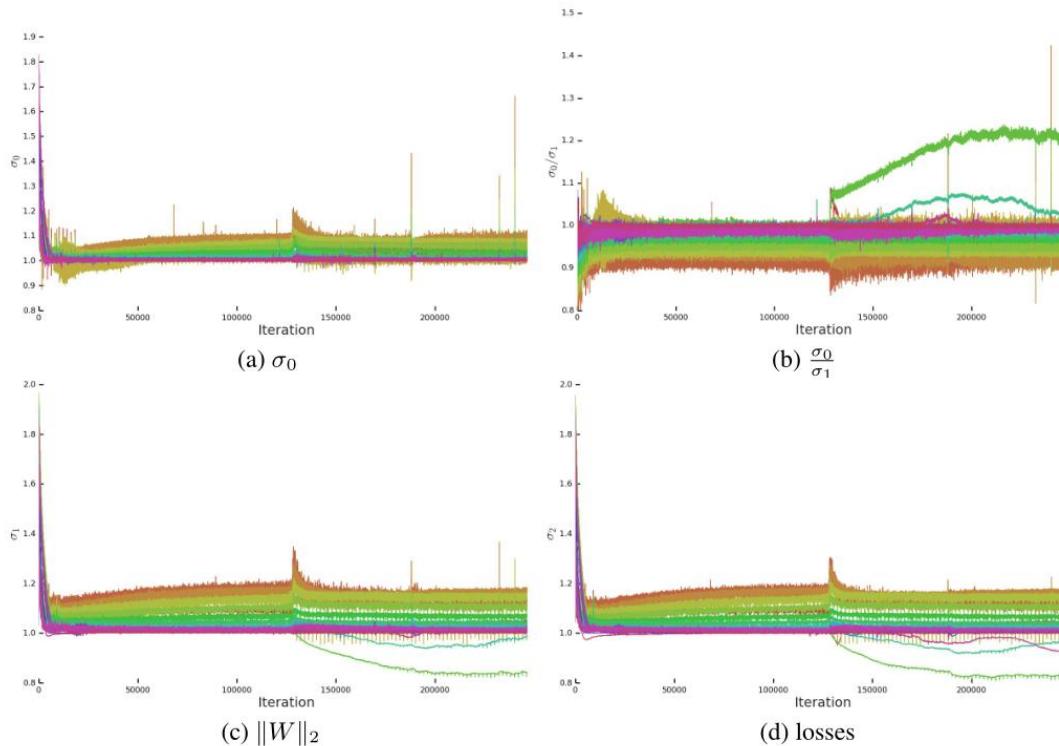


图 20: G 训练统计数据, G 中的 σ_0 正则化为 1。在 125000 次迭代后发生崩溃。

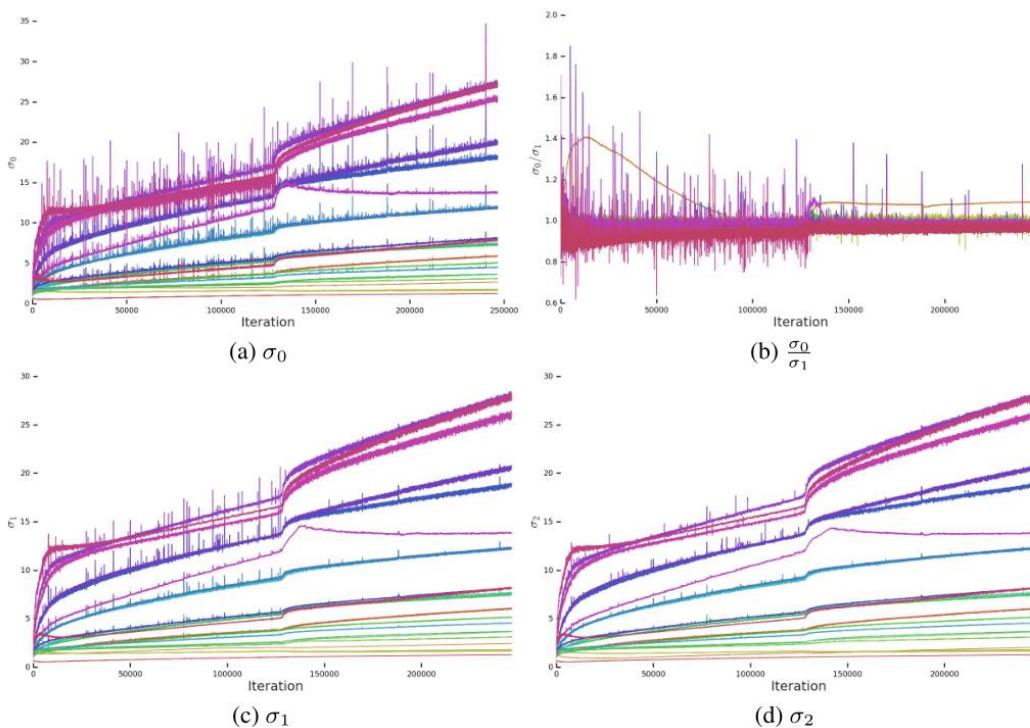


图 21: D 训练统计数据, G 中的 σ_0 正则化为 1。在 125000 次迭代后发生崩溃。

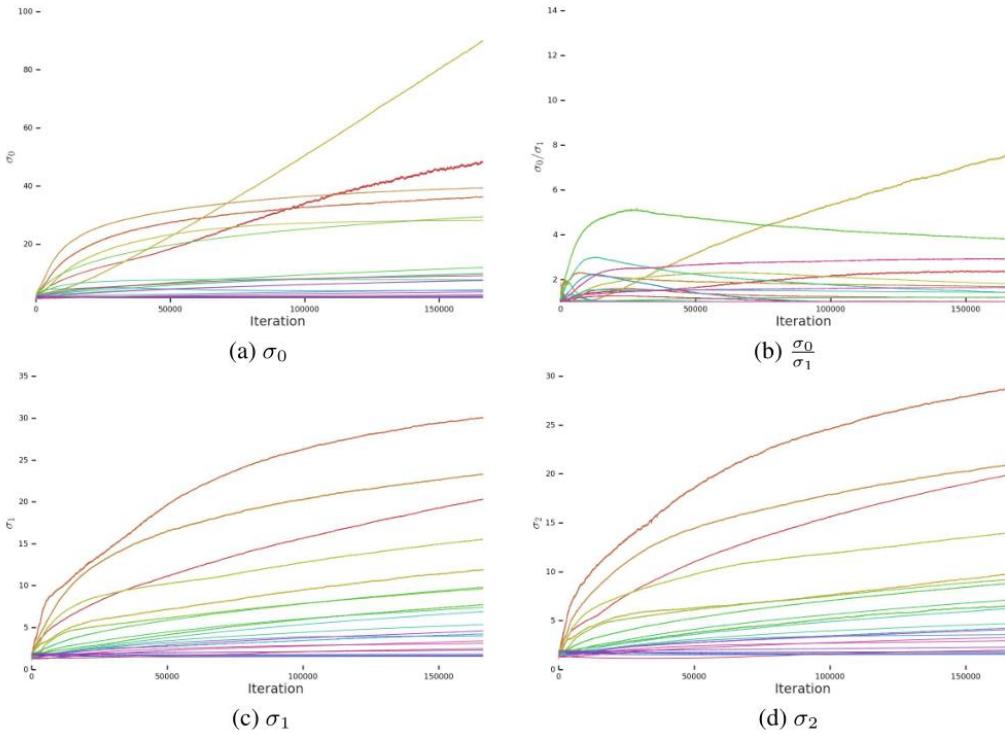


图 22：在 D 上使用 R1 梯度惩罚强度为 10 的 G 训练统计数据。此模型不会崩溃，但仅达到最大值 IS 为 55。

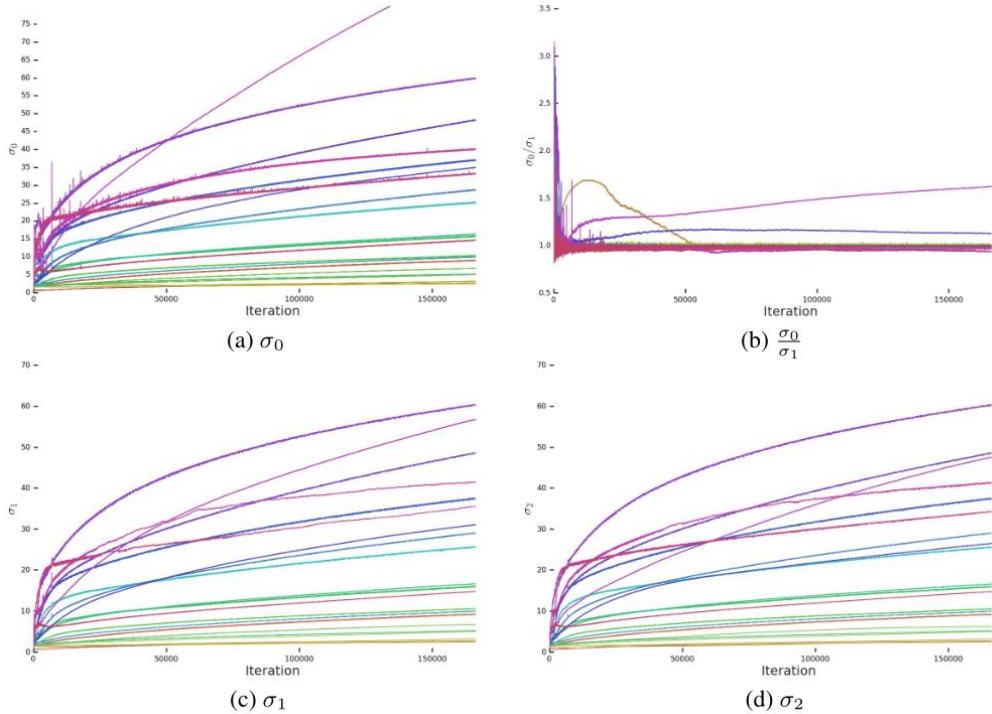


图 23：在 D 上使用 R1 梯度惩罚强度为 10 的 D 训练统计数据。此模型不会崩溃，但只会达到最大 IS 为 55。

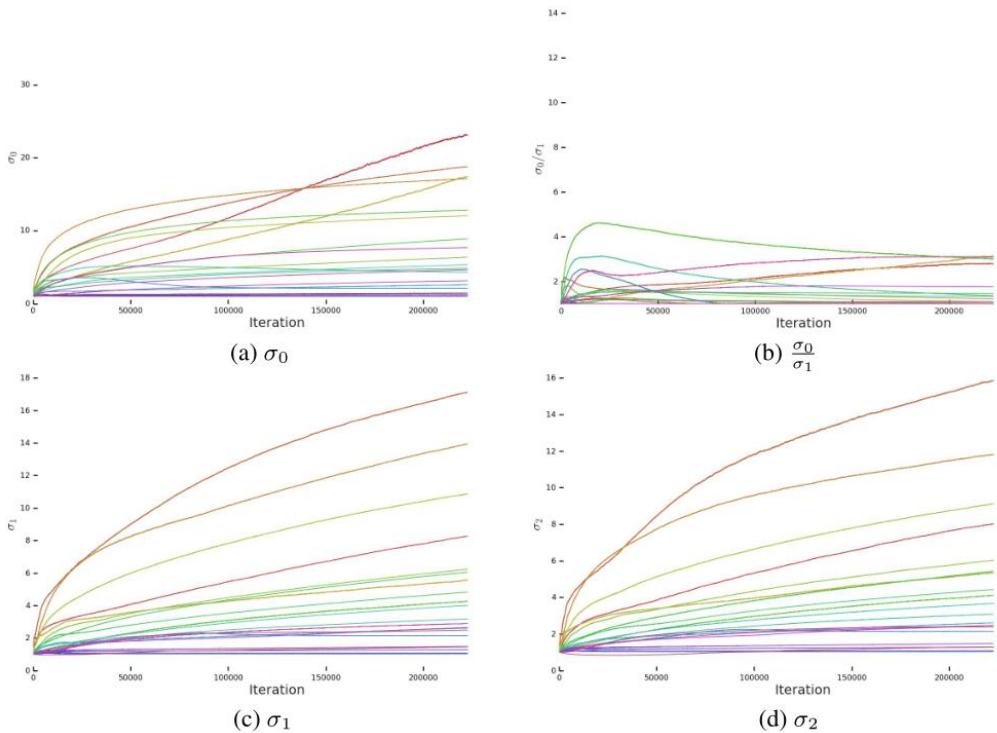


图 24：使用 Dropout（保持概率 0.8）的 G 训练统计数据应用于 D 的最后一个要素层。此模型不会崩溃，但仅达到最大值 IS 为 70。

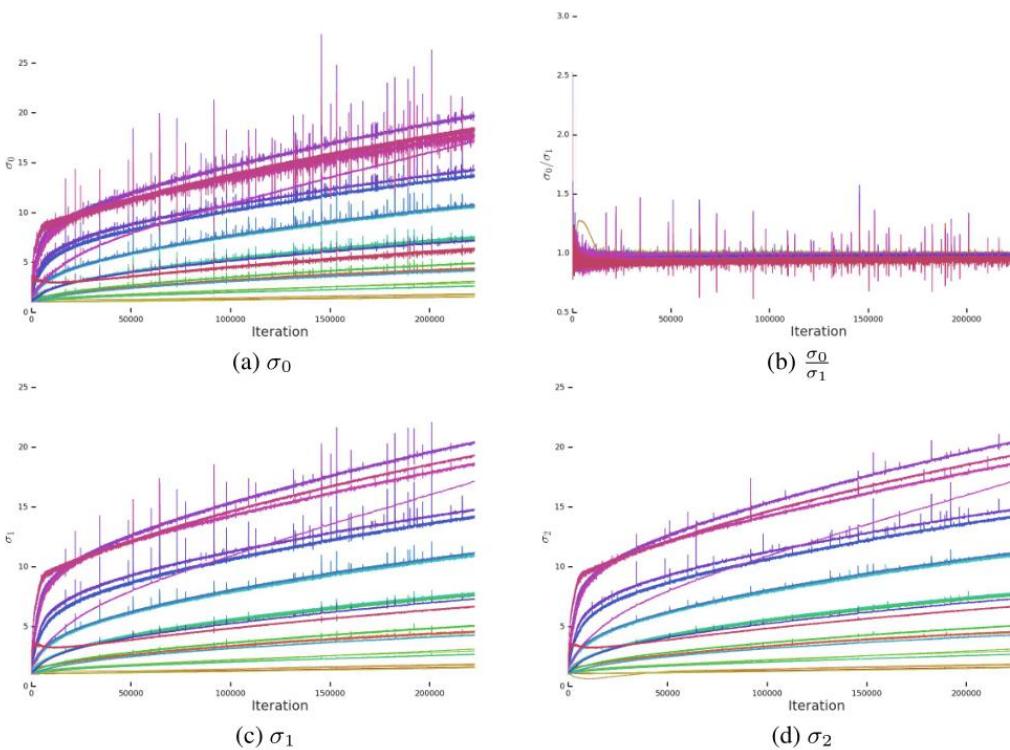


图 25：使用 Dropout (保持概率 0.8) 的 D 训练统计应用于 D 的最后一个要素层。此模型不会崩溃，但只达到最大值 IS 为 70。

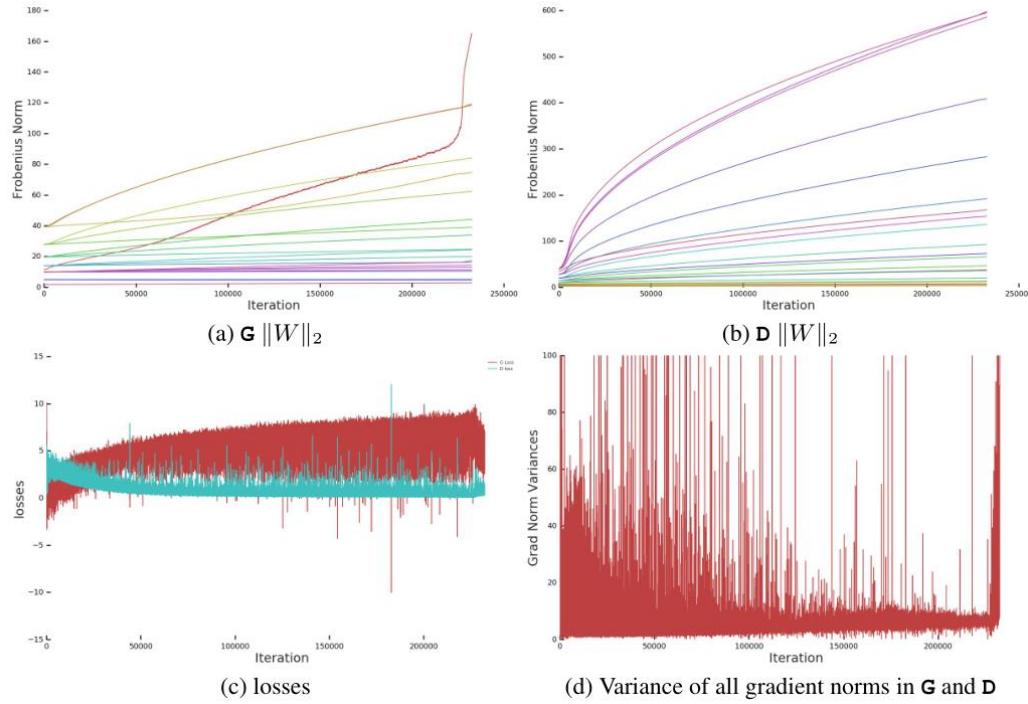


图 26：没有特殊修改的典型模型的附加培训统计数据。在 200000 次迭代后发生折叠。

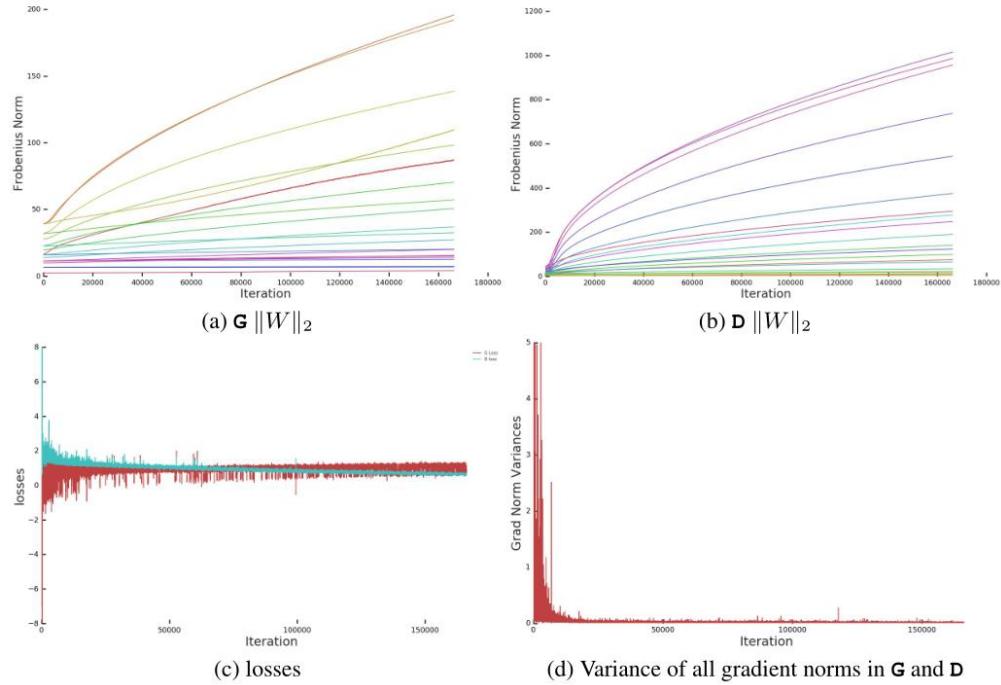


图 27：在 \mathbf{D} 上强度为 10 的 R1 梯度惩罚的附加训练统计数据。此模型不会崩溃，但只达到最大值 IS 为 55。

附件 G 差的结果

我们探索了一系列新颖的和现有的技术，这些技术最终会降低或影响我们环境中的性能。 我们在这里报告；我们对此部分的评估不像主要架构选择那样彻底。

- 我们发现将深度加倍（通过在每个上采样块或下采样块之后插入额外的残余块）会妨碍性能。
- 我们尝试在 G 和 D 之间共享类嵌入（而不是仅仅在 G 中）。这是通过将 G 的类嵌入替换为 G 嵌入的投影来实现的，就像在 G 的 BatchNorm 层中所做的那样。在我们最初的实际中，这似乎有助于加速训练，但我们发现这个技巧变得很差，并且对优化超参数敏感，特别是每 G 步的 D 步数选择。
- 我们尝试用 WeightNorm 替换 G 中的 BatchNorm (Salimans & Kingma, 2016)，但这种残缺训练。我们还试图删除 BatchNorm 并且只进行频谱规范化，但这也削弱了训练。
- 除了 Spectral Normalization 之外，我们尝试将 BatchNorm 添加到 D（包括类条件和无条件），但是这种削弱了训练。
- 我们尝试改变 G 和 D 中注意块的位置选择（并以不同的分辨率插入多个注意块），但发现在 128×128 这样做没有明显的好处，并且计算和内存成本显着增加。我们发现当移动到 256×256 时将注意块向上移动一个阶段有一个好处，这符合我们对分辨率提高的预期。
- 我们尝试在 G 或 D 或两者中使用 5 或 7 而不是 3 的过滤器大小。我们发现，在 G 中使用 5 的过滤器尺寸仅比基线提供了小的改进，但是计算成本不合理。所有其他设置降低了性能。
- 我们尝试在 128×128 处改变 G 和 D 中的卷积滤波器的扩张，但发现即使在任一网络中的少量扩张都会降低性能。
- 我们尝试用 G 中的双线性上采样来代替最近邻居的上采样，但这降低了性能。
- 在我们的一些模型中，我们观察到类条件模式崩溃，其中模型仅为类的子集输出一个或两个样本，但仍然能够为所有其他类生成样本。我们注意到崩溃的类具有相对于其他嵌入而变得非常大的嵌入，并且试图通过仅将权重衰减应用于共享嵌入来改善该问题。我们发现少量的重量衰减 (10^{-6}) 反而降低了性能，并且只有更小的值 (10^{-8}) 不会降低性能，但是这些值也太小而不能防止类向量爆炸。较高分辨率的模型似乎对这个问题更具弹性，而且我们的最终模型似乎都没有遭受这种类型的崩溃。
- 我们尝试使用 MLP 而不是从 G 类嵌入到其 BatchNorm 增益和偏差的线性投影，但没有发现这样做有任何好处。我们还通过 Spectrally Normalizing 这些 MLP 进行了实验，并提供了这些（和线性投影）的输出偏差，但没有注意到任何好处。
- 我们尝试了梯度范数裁剪（通常用于循环网络的全局变体，以及基于每个参数确定裁剪值的本地版本），但发现这并没有减轻不稳定性。

附件 H 超参数

我们在这项工作中进行了各种超参数扫描：

- 我们通过 $[10^{-5}, 5 \cdot 10^{-5}, 10^{-4}, 2 \cdot 10^{-4}, 4 \cdot 10^{-4}, 8 \cdot 10^{-4}, 10^{-3}]$ 扫描了每个网络学习率的笛卡尔积，并且最初发现了 SAGAN 设置（G 的学习率 10^{-4} , D 的学习率 $4 \cdot 10^{-4}$ ）在较低的批量大小时是最佳的；我们没有在更高的批量大小重复此扫描，但确实尝试将学习率减半并加倍，达到用于我们实验的减半设置。
- 我们通过 $[10^{-3}, 10^{-2}, 10^{-1}, 0.5, 1, 2, 3, 5, 10]$ 扫过 R1 梯度罚分强度。我们发现惩罚的强度与表现有负相关，但是 0.5 以上的设置赋予训练稳定性。
- 我们在 D 的最后一层通过 $[0.5, 0.6, 0.7, 0.8, 0.9, 0.95]$ 扫描了 DropOut 的保持概率。我们发现 DropOut 具有与 R1 相似的稳定效果，但也会降低性能。
- 我们通过 $[0.1, 0.2, 0.3, 0.4, 0.5]$ 扫描 D 的 Adam β_1 参数，发现它具有类似于 DropOut 的光正则化效果，但不能显着改善结果。任何一个残缺网络训练都会提高 β_1 个级数。
- 我们在 G 到 $[10^{-5}, 5 \cdot 10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}, 10^{-2}]$ 中选择了修正的正交正则化惩罚的强度，并且选择了 10^{-4} 。