Glow: 生成流
具有可逆的 1×1 卷积

Diederik P. Kingma*, Prafulla Dhariwal
OpenAI, San Francisco

摘 要

基于流的生成模型（Dinh 等，2014）在概念上具有吸引力，因为精确对数似然的易处理性，精确潜变量推断的易处理性以及训练和合成的可并行性。 在本文中，我们提出了 Glow，一种使用可逆 1×1 卷积的简单类型的生成流。 我们的方法，体现出了标准基准测试中对数似然性的显著改进。 也许最引人注目的是，我们证明了针对普通对数似然目标优化的生成模型能够实现高效逼真的合成和大图像处理。 我们的模型代码可以在下述地址找到 https://github.com/openai/glow.

1 介绍

机器学习领域的两个主要未解决的问题是：（1）数据效率：从少数数据点学习的能力，如人类;（2）泛化：对任务或其上下文变化的鲁棒性。 例如，当给定的输入与其训练分布不同时，AI 系统通常根本不起作用。 生成模型，是机器学习的一个主要分支，

*平等的贡献.

进行中的工作：.

图 1：从我们的模型中采样的合成名人; 有关架构和方法，请参阅第 3 节;有关更多结果，请参阅第 5 节。

是通过以下方式去克服上述限制：（1）学习现实世界模型，可能允许代理人在与世界实际互动之前计划世界模型，以及（2）学习有意义的输入特征，同时几乎不需要人工监督或标记。由于这些特征可以从大的未标记数据集中学习，并且不一定是特定于任务的，因此基于这些特征的下游解决方案可能更健壮，数据效率更高。在本文中，除了中间应用之外，我们致力于实现这一最终愿景，旨在改进生成模型的最新技术。

生成建模通常涉及在非常高维的输入数据内建模的极具挑战性的任务，通常以完整的联合概率分布的形式指定。由于这种联合模型可能捕获数据中存在的所有模式，因此精确生成模型的应用几乎是无穷无尽的。即时应用程序与语音合成，文本分析，半监督学习和基于模型的控制一样多种多样;请参阅第 4 节以获取参考。

近年来，生成建模学科在能力方面取得了巨大的飞跃，主要采用基于可能性的方法（Graves，2013; Kingma 和 Welling，2013,2018; Dinh 等，2014; van den Oord 等，2016a）和生成对抗网络（GAN）（Goodfellow 等，2014）（见第 4 节）。基于似然的方法可分为三类:

1. 自回归模型（Hochreiter 和 Schmidhuber，1997; Graves，2013; van den Oord 等，2016a，b; Van Den Oord 等，2016）。 那些具有简单的优点，但具有合成具有有限的可并行性的缺点，因为合成的计算长度与数据的维度成比例; 这对于大型图像或视频来说尤其麻烦。
2. 变分自动编码器（VAE）（Kingma 和 Welling，2013,2018），它们优化了数据对数似然的下界。 变分自动编码器具有训练和合成的可并行性的优点，但优化相对具有挑战性（Kingma 等，2016）。
3. 基于流的生成模型，首先在 NICE（Dinh 等人，2014）中描述并在 RealNVP 中延伸（Dinh 等人，2016）。 我们将在以下部分中解释这类模型背后的关键思想。

与 GAN（Goodfellow 等，2014）和 VAE（Kingma 和 Welling，2013）相比，基于流的生成模型迄今为止在研究界很少受到关注。 基于流的生成模型的一些优点包括:

- 精确的潜变量推断和对数似然评估。在 VAE 中，人们只能推断出与数据点相对应的潜在变量的值。 GAN 根本没有编码器来推断潜伏者。在可逆的生成模型中，这可以在没有近似的情况下完全完成。这不仅可以实现准确的推理，还可以优化数据的精确对数似然，而不是其下限。

- 高效的推理和有效的合成。自回归模型，例如 Pixel-CNN（van den Oord 等，2016b）也是可逆的，但是来自这些模型的合成难以并行化，并且通常在并行硬件上效率低。像 Glow（和 RealNVP）这样的基于流的生成模型可以有效地进行推理和合成的并行化。

- 下游任务的有用潜在空间。自回归模型的隐藏层具有未知的边际分布，使得执行有效的数据操作变得更加困难。在 GAN 中，数据点通常不能直接在潜在空间中表示，因为它们没有编码器，并且可能没有对数据分布的完全支持。 （Grover 等，2018）。对于可逆生成模型和 VAE 而言，情况并非如此，后者允许各种应用，例如数据点之间的插值和现有数据点的有意义修改。

- 节省内存的巨大潜力。在可逆神经网络中计算梯度需要一定量的内存，而不是线性的深度，如 RevNet 论文（Gomez 等，2017）所述。.

在本文中，我们提出了一个新的生成流 Glow，其中包含第 3 节中描述的各种新元素。在第 5 节中，我们将模型与之前的流进行定量比较，在第 6 节中，我们研究了模型的定性方面。 高分辨率数据集。

2 背景：基于流的生成模型

设 x 是具有未知真实分布 x~p*(x)的高维随机向量。 我们收集了 i.i.d. 数据集 D，并选择带参数θ的模型 Pθ(x)。 在离散数据 x 的情况下，对数似然目标等同于最小化：

$$\mathcal{L}(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N} -\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \tag{1}$$

在连续数据 x 的情况下，我们最小化以下内容:

$$\mathcal{L}(\mathcal{D}) \simeq \frac{1}{N}\sum_{i=1}^{N} -\log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}^{(i)}) + c \tag{2}$$

其中 $\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} + u$，u~U(0,a)，c = -M·log a，其中 a 由数据的离散化水平确定，M 是 x 的维数。 两个目标（方程（1）和（2））以 nat 或比特来衡量预期的压缩成本; 见（Dinh 等，2016）。 通过使用微型数据的随机梯度下降来完成优化（Kingma 和 Ba，2015）。

在大多数基于流的生成模型中（Dinh 等，2014,2016），生成过程被定义为:

$$\mathbf{z} \sim p_{\boldsymbol{\theta}}(\mathbf{z}) \tag{3}$$

$$\mathbf{x} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z}) \tag{4}$$

其中 z 是潜变量，Pθ(z)具有（通常简单的）易处理密度，例如球形多变量高斯分布：Pθ(z)= N(Z;0,I)。 函数 gθ(..)是可逆的，也称为双射的，使得给定数据点 x，潜在变量推断由 $\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{g}_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$完成。 为简洁起见，我们将省略 f 和 g 的下标。

我们关注的函数是 f（以及同样地，g）由一系列变换组成：f=f1∘f2∘...∘fK，这样 x 和 z 之间的关系可以写成：

$$\mathbf{x} \xleftrightarrow{\mathbf{f}_1} \mathbf{h}_1 \xleftrightarrow{\mathbf{f}_2} \mathbf{h}_2 \cdots \xleftrightarrow{\mathbf{f}_K} \mathbf{z} \tag{5}$$

这种可逆转换序列也称为（归一化）流（Rezende 和 Mohamed，2015）。 在等式变量的变化下。 （4）给定数据点的模型的概率密度函数（pdf）可写为:

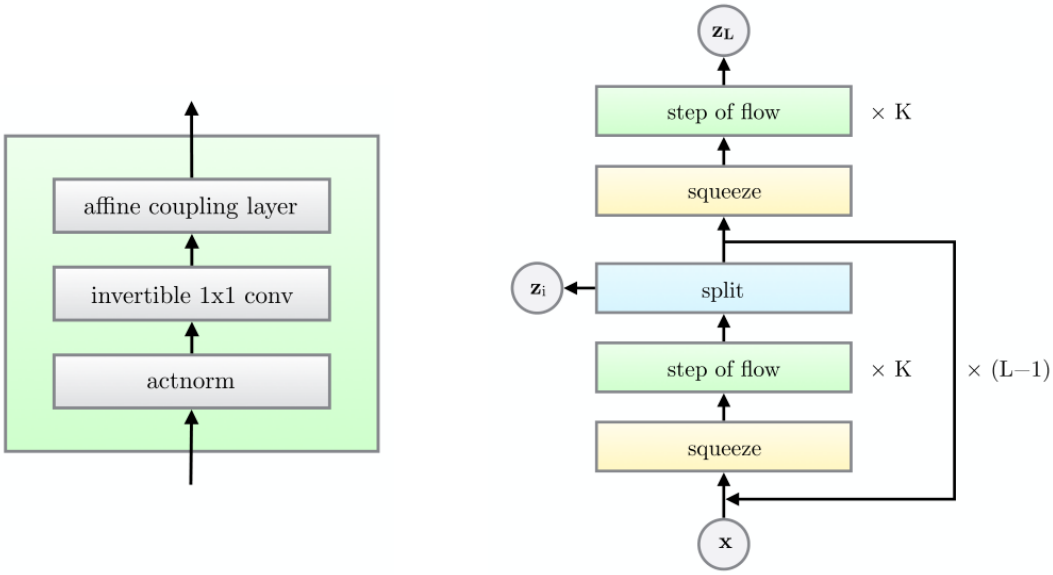
$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log p_{\boldsymbol{\theta}}(\mathbf{z}) + \log|\det(d\mathbf{z}/d\mathbf{x})| \tag{6}$$

$$= \log p_{\boldsymbol{\theta}}(\mathbf{z}) + \sum_{i=1}^{K} \log|\det(d\mathbf{h}_i/d\mathbf{h}_{i-1})| \tag{7}$$

我们定义 h0 $\triangleq$ x 和 hk $\triangleq$ z 以简洁表示。 标量值 $\log|\det(d\mathbf{h}_i/d\mathbf{h}_{i-1})|$是雅可比矩阵$(d\mathbf{h}_i/d\mathbf{h}_{i-1})$的行列式的绝对值的对数，也称为对数行列式。 该值是在转换 fi 下从 hi-1 到 hi 的对数密度的变化。 虽然它可能看起来令人生畏，但它的价值可以非常简单地计算某些变换选择，如先前在（Deco 和 Brauer，1995; Dinh 等，2014; Rezende 和 Mohamed，2015; Kingma 等，2016;）。 基本思想是选择 Jacobian$(d\mathbf{h}_i/d\mathbf{h}_{i-1})$是三角矩阵的变换。 对于那些转换，log-determinant 很简单:

$$\log|\det(d\mathbf{h}_i/d\mathbf{h}_{i-1})| = \mathtt{sum}(\log|\mathtt{diag}(d\mathbf{h}_i/d\mathbf{h}_{i-1})|) \tag{8}$$

其中 sum()取所有向量元素的总和，log()采用元素对数，diag()采用雅可比矩阵的对角线。



(a) One step of our flow. (b) Multi-scale architecture (Dinh et al., 2016).

图 2：我们提出了一个生成流，其中每个步骤（左）由一个 actnorm 步骤组成，然后是可逆的 1×1 卷积，然后是仿射变换（Dinh 等，2014）。 该流与多尺度架构（右）相结合。 见第 3 节和表 1。

表 1：我们提出的流的三个主要组成部分，它们的反转以及它们的对数决定因素。 这里，x 表示图层的输入，y 表示其输出。 x 和 y 都是具有空间尺寸（h; w）和通道尺寸 c 的形状[h×w×c]的张量。 使用（i; j），我们将空间索引表示为张量 x 和 y。 函数 NN（）是非线性映射，例如 ResNets（He 等人，2016）和 RealNVP（Dinh 等人，2016）中的（浅）卷积神经网络。

Description	Function	Reverse Function	Log-determinant
Actnorm. See Section 3.1	$\forall i,j: \mathbf{y}_{i,j} = \mathbf{s} \odot \mathbf{x}_{i,j} + \mathbf{b}$	$\forall i,j: \mathbf{x}_{i,j} = (\mathbf{y}_{i,j} - \mathbf{b})/\mathbf{s}$	$h \cdot w \cdot \mathtt{sum}(\log \lvert\mathbf{s}\rvert)$
Invertible 1×1 convolution. $\mathbf{W}: [c \times c]$. See Section 3.2	$\forall i,j: \mathbf{y}_{i,j} = \mathbf{W}\mathbf{x}_{i,j}$	$\forall i,j: \mathbf{x}_{i,j} = \mathbf{W}^{-1}\mathbf{y}_{i,j}$	$h \cdot w \cdot \log \lvert \det(\mathbf{W}) \rvert$ or $h \cdot w \cdot \mathtt{sum}(\log \lvert\mathbf{s}\rvert)$ (see eq. (10))
Affine coupling layer. See Section 3.3 and (Dinh et al., 2014)	$\mathbf{x}_a, \mathbf{x}_b = \mathtt{split}(\mathbf{x})$ $(\log \mathbf{s}, \mathbf{t}) = \mathtt{NN}(\mathbf{x}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{y}_a = \mathbf{s} \odot \mathbf{x}_a + \mathbf{t}$ $\mathbf{y}_b = \mathbf{x}_b$ $\mathbf{y} = \mathtt{concat}(\mathbf{y}_a, \mathbf{y}_b)$	$\mathbf{y}_a, \mathbf{y}_b = \mathtt{split}(\mathbf{y})$ $(\log \mathbf{s}, \mathbf{t}) = \mathtt{NN}(\mathbf{y}_b)$ $\mathbf{s} = \exp(\log \mathbf{s})$ $\mathbf{x}_a = (\mathbf{y}_a - \mathbf{t})/\mathbf{s}$ $\mathbf{x}_b = \mathbf{y}_b$ $\mathbf{x} = \mathtt{concat}(\mathbf{x}_a, \mathbf{x}_b)$	$\mathtt{sum}(\log(\lvert\mathbf{s}\rvert))$

3 提出的生成流

我们提出了一个新的流，建立在 NICE 和 RealNVP 流的基础上（Dinh et al。，2014,2016）。它由一系列流组成，结合在一个多尺度的架构中；参见图 2.流的每个步骤包括 actnorm（第 3.1 节），然后是可逆的 1×1 卷积（第 3.2 节），接着是耦合层（第 3.3 节）。

该流与多尺度架构相结合；由于空间限制，我们参考（Dinh 等，2016）了解更多细节。 该架构具有流动深度 K 和层数 L（图 2）。.

3.1 Actnorm：具有数据相关初始化的缩放和偏置层

在 Dinh 等人（2016），作者提出使用批量标准化（Ioffe 和 Szegedy，2015）来缓解训练深度模型时遇到的问题。 但是，因为方差

通过批量归一化添加的激活噪声与每 GPU 或其他处理单元（PU）的小批量大小成反比，已知性能对于小的每 PU 小批量大小而降低。 对于大图像，由于内存限制，我们学习每个 PU 的小批量大小 1。我们提出了一个 actnorm 层（用于激活标准化），它使用每个通道的标度和偏差参数执行激活的仿射变换，类似于批量标准化。 初始化这些参数，使得在给定初始数据小批量的情况下，每个通道的后行为动作具有零均值和单位方差。 这是一种依赖于数据的初始化（Salimans 和 Kingma，2016）。初始化后，比例和偏差被视为与数据无关的常规可训练参数。

3.2 可逆的 1×1 卷积

（Dinh 等人，2014 年，2016 年）提出了一个包含相当于排列的流，该排列反转了通道的排序。我们建议用（学习的）可逆 1×1 卷积替换该固定置换，其中权重矩阵被初始化为随机旋转矩阵。 注意，具有相等数量的输入和输出通道的 1×1 卷积是置换操作的概括。

具有 c× c 权矩阵 W 的 h × w × c 张量 h 的可逆 1×1 卷积的对数行列式很容易计算:

$$\log\left|\det\left(\frac{d\,\mathtt{conv2D}(\mathbf{h};\mathbf{W})}{d\,\mathbf{h}}\right)\right| = h\cdot w\cdot\log|\det(\mathbf{W})| \tag{9}$$

计算或区分 det（W）的成本是 O（c3），其通常与成本计算 conv2D（h; W），即 O(h·w·c2)相当。 我们将权重 W 初始化为随机旋转矩阵，其对数行列式为 0; 在一个 SGD 步骤之后，这些值开始偏离 0。

LU 分解 通过在 LU 分解中直接参数化 W，可以将计算 det(W)的成本从 O(c3)减少到 O(c)：

$$W = PL(U + \mathrm{diag}(s)) \tag{10}$$

其中 P 是置换矩阵，L 是下三角矩阵，对角线上有一个，U 是对角线上有零的上三角矩阵，s 是矢量。 那么对数决定因素就是：

$$\log|\det(W)| = \mathrm{sum}(\log|s|) \tag{11}$$

对于大型 c，计算成本的差异将变得很大，尽管对于我们实验中的网络，我们没有测量壁钟计算时间的巨大差异。

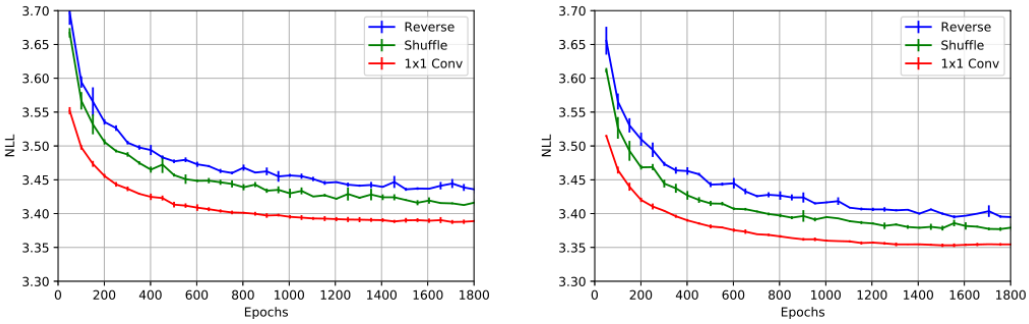
在该参数化中，我们通过首先对随机旋转矩阵 W 进行采样来初始化参数，然后计算 P 的对应值（其保持固定）以及 L 和 U 和 s 的相应初始值（其被优化）。

3.3 仿射耦合层

其中前向函数，反向函数和对数行列式在计算上有效的强大的可逆转换是(Dinh 等人 2014,2016)中引入的仿射耦合层。参见表 1 添加剂耦合层是特殊情况，其中 s=1 且对数行列式为 0。

零初始化 我们用零初始化每个 NN（）的最后一个卷积，使得每个仿射耦合层最初执行一个同一性函数;我们发现这有助于培训非常深入的网络。

拆分和连接 如（Dinh 等，2014）所述，split（）函数将输入张量沿通道维度分成两半，而 concat（）操作执行相应的反向操作：连接成单个张量。在（Dinh 等，2016）中，引入了另一种类型的分裂：沿着使用棋盘图案的空间维度。在这项工作中，我们只沿通道维度执行拆分，简化了整体架构。



(a) Additive coupling. (b) Affine coupling.

图 3：三种变体的比较 - RealNVP 中描述的反转操作，固定随机排列，以及我们提出的可逆 1×1 卷积，具有加性（左）与仿射（右）耦合层。 我们用不同的随机种子绘制三次运行的平均值和标准差。

排列 上面的每个流程步骤之前都应该对变量进行某种排列，以确保在充分的流程步骤之后，每个维度都可以影响其他每个维度。 在（Dinh 等人，2014,2016）中具体完成的排列类型等同于在执行附加耦合层之前简单地反转通道（特征）的排序。 另一种方法是执行（固定）随机排列。 我们的可逆 1x1 卷积是这种排列的推广。 在实验中，我们比较了这三种选择。

4 相关工作

这项工作建立在（Dinh 等，2014）（NICE）和（Dinh 等，2016）（RealNVP）中提出的想法和流程的基础上;在整篇文章中都与这项工作进行了比较。在（Papamakarios et al。，2017）（MAF）中，作者提出了基于 IAF 的生成流（Kingma 等，2016）;然而，由于 MAF 的合成是不可并行化的，因此效率低，我们在比较中省略了它。自回归（AR）模型的合成（Hochreiter 和 Schmidhuber，1997; Graves，2013; van den Oord 等，2016a，b; Van Den Oord 等，2016）同样是不可并行化的。使用 AR 模型，高维数据的合成通常需要多个数量级;见（Kingma et al。，2016; Oord et al。，2017）作为证据。使用我们最大的型号采样 256×256 张图像在当前硬件上只需不到一秒钟。

GANs（Goodfellow et al。，2014）可以说是因其合成大而逼真的图像的能力而闻名（Karras 等，2017），与基于可能性的方法形成对比。 GAN 的缺点是它们普遍缺乏潜在空间编码器，它们普遍缺乏对数据的全面支持（Grover et al。，2018），它们的优化难度以及评估过度拟合和泛化的难度。

5 定量实验

我们通过比较我们的新流与 RealNVP 的比较来开始我们的实验（Dinh 等，2016）。然后，我们将模型应用于其他标准数据集，并将对数似然性与先前的生成模型进行比较。有关优化详细信息，请参阅附录。 在我们的实验中，我们让每个 NN（）都有三个卷积层，其中两个隐藏层具有 ReLU 激活函数和 512 个通道。 第一个和最后一个卷积是 3×3，而中心卷积是 1×1，因为它的输入和输出都有大量的通道，与第一个和最后一个卷积相反。.

[2]更具体地说，在批量大小为 1 时生成 256×256 图像在单个 1080 Ti 上需要大约 130 毫秒，在 K80 上大约需要 550 毫秒

表 2：与 RealNVP 相比，我们模型的每维度比特的最佳结果.

Model	CIFAR-10	ImageNet 32x32	ImageNet 64x64	LSUN (bedroom)	LSUN (tower)	LSUN (church outdoor)
RealNVP	3:49	4:28	3:98	2.72	2.81	3.08
Glow	3:35	4:09	3:81	2:38	2:46	2:67

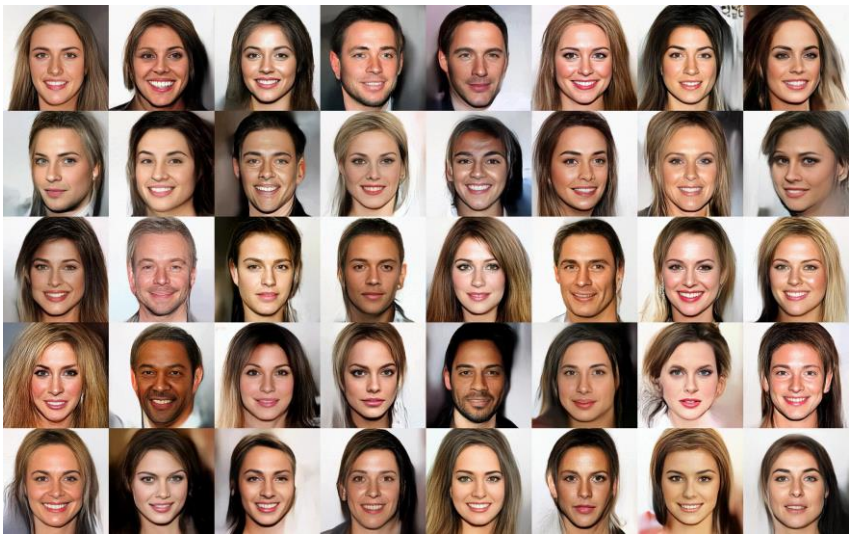


图 4：来自模型的随机样本，温度为 0.7

使用可逆 1×1 卷积获得收益 我们选择第 3 节中描述的体系结构，并考虑通道变量置换的三种变体 - RealNVP 中描述的反转操作，固定随机置换和我们的可逆 1×1 卷积。我们比较了仅具有附加耦合层的模型和具有仿射耦合的模型。如前所述，我们使用依赖于数据的初始化初始化所有模型，该初始化规范化每个层的激活。所有模型都经过训练，K = 32 且 L = 3.具有 1×1 卷积的模型具有可忽略的 0.2%的参数量。

我们比较了 CIFAR-10（Krizhevsky，2009）数据集的平均负对数似然（每维度的比特），保持所有训练条件不变并对三个随机种子求平均值。结果如图 3 所示。正如我们所看到的，对于加性和仿射耦合，可逆 1×1 卷积实现了较低的负对数似然并且收敛更快。仿射耦合模型也比加性耦合模型收敛得更快。我们注意到可逆 1×1 卷积模型的挂钟时间的增加仅为 7%，因此该操作也是计算上有效的。

与 RealNVP 在标准基准测试中的比较 除了置换操作之外，RealNVP 架构还有其他差异，例如空间耦合层。为了验证我们提出的架构与 RealNVP 架构的整体竞争力，我们在各种自然图像数据集上比较我们的模型。特别是，我们比较了 CIFAR-10，ImageNet（Russakovsky 等，2015）和 LSUN（Yu 等，2015）数据集。我们遵循与（Dinh 等，2016）相同的预处理。对于 Imagenet，我们使用 ImageNet 的 32×32 和 64×64 下采样版本（Oord et al.，2016），对于 LSUN，我们下采样到 96×96 并采用 64×64 的随机作物。我们还包括我们模型的比特/维度在我们的定性实验中使用 256×256 CelebA HQ 进行培训。如表 2 所示，我们的模型在所有数据集上实现了显著改进。

[3]由于原始 CelebA HQ 数据集没有验证集，我们将其分为 27000 个图像的训练集和 3000 个图像的验证集

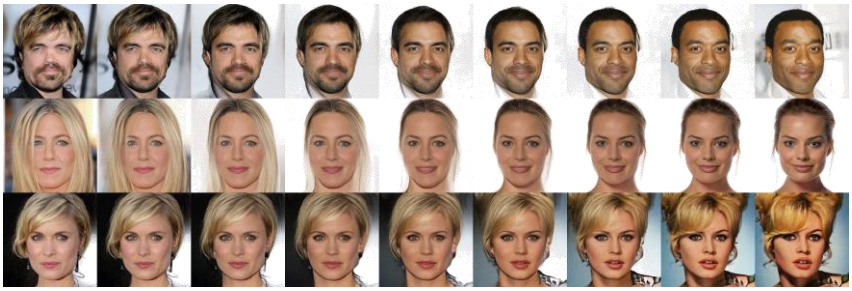


图 5：实际图像之间潜在空间中的线性插值

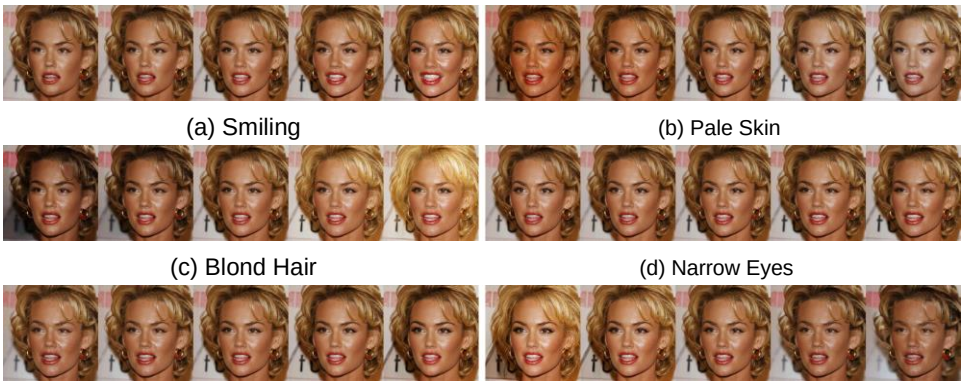


图 6：操纵面部属性。 每行是通过沿对应于属性的向量内插图像的潜码来制作的，中间图像是原始图像

6 定性实验

我们现在研究模型在高分辨率数据集上的定性方面。我们选择 CelebA-HQ 数据集（Karras 等，2017），它由来自 CelebA 数据集的 30000 个高分辨率图像组成，并训练与上面相同的结构，但现在用于分辨率为 2562 的图像，

K = 32 和 L = 6.为了以略微降低色彩保真度为代价提高视觉质量，我们在 5 位图像上训练我们的模型。我们的目标是研究我们的模型是否可以扩展到高分辨率，生成逼真的样本，并产生有意义的潜在空间。由于设备内存的限制，在这些分辨率下，我们使用每个 PU 的小批量大小 1，并使用渐变检查点（Salimans 和 Bulatov，2017）。在未来，我们可以通过利用模型的可逆性来使用与深度无关的恒定内存量（Gomez 等，2017）。

与早期基于可能性的生成模型的研究一致（Parmar 等，2018），我们发现从降温模型中采样通常会产生更高质量的样本。当用温度 T 采样时，我们从分布$p_{\boldsymbol{\theta},T}(\mathbf{x}) \propto (p_{\boldsymbol{\theta}}(\mathbf{x}))^{T^2}$中采样。在添加剂耦合层的情况下，这可以简单地通过将$p_{\boldsymbol{\theta}}(\mathbf{z})$的标准偏差乘以因子 T 来实现。

合成与插值 图 4 显示了从我们的模型中获得的随机样本。对于基于非自回归似然的模型，图像具有极高的质量。为了了解我们如何插值，我们采用一对真实图像，用编码器对它们进行编码，并在潜伏期之间进行线性插值以获得样本。图 5 中的结果表明，发生器分布的图像流形非常平滑，几乎所有中间样本看起来都像真实的面。

语义操纵 我们现在考虑修改图像的属性。为此，我们使用 CelebA 数据集中的标签。每个图像都有一个二进制标签，对应于微笑，金发，年轻等属性的存在与否。这为每个属性提供了 30000 个二进制标签。

图 7：来自 5 位 LSUN 卧室的模型样品，温度为 0.875。 第 64,96 和 128 号决议 [4]

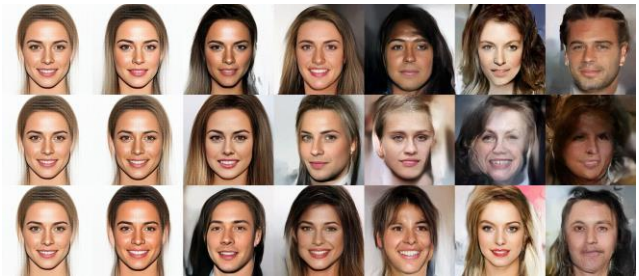


图 8：温度变化的影响。 从左到右，在温度下获得样品 0, 0.25, 0.6, 0.7, 0.8, 0.9, 1.0

然后，我们计算具有属性的图像的平均潜在矢量 Zpos 和没有的图像的 Zneg，然后使用差异（Zpos-Zneg）作为操纵的方向。 请注意，这是一个相对较少的监督，并且是在训练模 型后进行的（训练时没有使用标签），这使得对各种不同的目标属性非常容易。 结果如图 6 所示。

温度和模型深度的影响 图 8 显示了样品质量和多样性如何随温度变化。 最高温度具有噪声图像，可能是由于过高估计数据分布的熵，因此我们选择 0.7 的温度作为样品的多样性和质量的最佳点。 图 9 显示了模型深度如何影响模型学习远程依赖关系的能力。

4 对于 128×128 和 96×96 版本，我们将中心裁剪为原始图像，然后进行下采样。 对于 64×64 版本，我们采用了来自 96×96 下采样图像的随机作物，如 Dinh 等人所做。（2016）

图 9：左侧浅模型与右侧深模型的样本。 浅模型具有 L = 4 级，而深模型具有 L = 6 级

7 结论

我们提出了一种新型流，创造了 Glow，并在标准图像建模基准上的对数似然性方面展示了改进的定量性能。 此外，我们证明了在高分辨率人脸训练时，我们的模型能够合成逼真的图像。 据我们所知，我们的模型是文献中第一个基于可能性的模型，可以有效地合成高分辨率的自然图像。

参考文献

Deco, G. and Brauer, W. (1995). Higher order statistical decorrelation without information loss. Advances in Neural Information Processing Systems, pages 247–254.

Dinh, L., Krueger, D., and Bengio, Y. (2014). Nice: non-linear independent components estimation. arXiv preprint arXiv:1410.8516.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using Real NVP. arXiv preprint arXiv:1605.08803.

Gomez, A. N., Ren, M., Urtasun, R., and Grosse, R. B. (2017). The reversible residual network: Backpropagation without storing activations. In Advances in Neural Information Processing Systems, pages 2211–2221.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680.

Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

Grover, A., Dhar, M., and Ermon, S. (2018). Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In AAAI Conference on Artificial Intelligence.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. arXiv preprint arXiv:1603.05027.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. Neural computation, 9(8):1735–1780.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. Proceedings of the International Confer-ence on Learning Representations 2015.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In Advances in Neural Information Processing Systems, pages 4743– 4751.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. Proceedings of the 2nd International Confer-ence on Learning Representations.

Kingma, D. P. and Welling, M. (2018). Variational autoencoders. Under Review.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

Oord, A. v. d., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. arXiv preprint arXiv:1601.06759.

Oord, A. v. d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G. v. d., Lockhart, E., Cobo, L. C., Stimberg, F., et al. (2017). Parallel wavenet: Fast high-fidelity speech synthesis. arXiv preprint arXiv:1711.10433.

Papamakarios, G., Murray, I., and Pavlakou, T. (2017). Masked autoregressive flow for density estimation. In Advances in Neural Information Processing Systems, pages 2335–2344.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., and Ku, A. (2018). Image transformer. arXiv preprint arXiv:1802.05751.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Proceedings of The 32nd International Conference on Machine Learning, pages 1530–1538.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3):211–252.

Salimans, T. and Bulatov, Y. (2017). Gradient checkpointing. https://github.com/openai/ gradient-checkpointing.

Salimans, T. and Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. arXiv preprint arXiv:1602.07868.

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. (2016a). Pixel recurrent neural networks. arXiv preprint arXiv:1601.06759.

van den Oord, A., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. (2016b). Conditional image generation with PixelCNN decoders. arXiv preprint arXiv:1606.05328.

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365.

A Additional quantitative results

See Table 3.

Table 3: Quantiative results in bits per dimension on the test set.

Dataset	Glow
CIFAR-10, 32 32, 5-bit	1.67
ImageNet, 32 32, 5-bit	1.99
ImageNet, 64 64, 5-bit	1.76
CelebA HQ, 256 256, 5-bit	1.03

B Simple python implementation of the invertible 1 1 convolution

```
# Invertible 1x1 conv
def invertible_1x1_conv(z, logdet, forward=True):
    # Shape
    h,w,c = z.shape[1:]

    # Sample a random orthogonal matrix to initialise weights w_init =
    np.linalg.qr(np.random.randn(c,c))[0]
    w = tf.get_variable("W", initializer=w_init)

    # Compute log determinant
    dlogdet = h * w * tf.log(abs(tf.matrix_determinant(w)))

    if forward:
        # Forward computation
        _w = tf.reshape(w, [1,1,c,c])
        z = tf.nn.conv2d(z, _w, [1,1,1,1], 'SAME')
        logdet += dlogdet
```

```
        return z, logdet
    else:
        # Reverse computation
        _w = tf.matrix_inverse(w)
        _w = tf.reshape(_w, [1,1,c,c])
        z = tf.nn.conv2d(z, _w, [1,1,1,1], 'SAME')
        logdet -= dlogdet

        return z, logdet
```

C Optimization details

We use the Adam optimizer (Kingma and Ba, 2015) with = 0:001 and default $_1$ and $_2$. In out quantitative experiments (Section 5, Table 2) we used the following hyperparameters (Table 4).

Table 4: Hyperparameters for results in Section 5, Table 2

Dataset	Minibatch Size	Levels (L)	Depth per level (K)	Coupling
CIFAR-10	512	3	32	Affine
ImageNet, 32 32	512	3	48	Affine
ImageNet, 64 64	128	4	48	Affine
LSUN, 64 64	128	4	48	Affine

In our qualitative experiments (Section 6), we used the following hyperparameters (Table 5)

Table 5: Hyperparameters for results in Section 6

Dataset	Minibatch Size	Levels (L)	Depth per level (K)	Coupling
LSUN, 64 64, 5-bit	128	4	48	Additive
LSUN, 96 96, 5-bit	320	5	64	Additive
LSUN, 128 128, 5-bit	160	5	64	Additive
CelebA HQ, 256 256, 5-bit	40	6	32	Additive

D Extra samples from qualitative experiments

For the class conditional CIFAR-10 and 32 32 ImageNet samples, we used the same hyperparameters as the quantitative experiments, but with a class dependent prior at the top-most level. We also added a classification loss to predict the class label from the second last layer of the encoder, with a weight of = 0:01. The results are in Figure 10.

E Extra samples from the quantitative experiments

For direct comparison with other work, datasets are preprocessed exactly as in Dinh et al. (2016). Results are in Figure 11 and Figure 12.

(a) Class conditional CIFAR-10 samples

(b) Class conditional 32 32 ImageNet samples

图 10：分别在 5 位 CIFAR-10 和 32×32 ImageNet 上的类条件样本。 温度 0.75

图 11：分别来自 8 位，64 64 个 LSUN 卧室，教堂和塔楼的样本。 温度 1.0

图 12：来自无条件模型的样品，其中仿射耦合层在温度为 1.0 的 CIFAR-10 数据集上训练。