MuseGAN：用于符号音乐生成和伴奏的多轨序列生成对抗网络

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, Yi-Hsuan Yang

1 中研院资讯科技创新研究中心，台北，台湾

2 国立清华大学计算机科学系，新竹，台湾

翻译：IDO，https://github.com/prodbyido

摘要

生成音乐与生成图像和视频有一些显著的区别。首先，音乐是一种时间上的艺术，需要一个时间模型。其次，音乐通常由多个乐器/轨道组成，具有自己的时间动态，但集体地它们相互依赖地随着时间展开。最后，在复调音乐中，音符通常被分组成和弦、琶音或旋律，因此引入音符的时间顺序并不自然适合。在本文中，我们提出了三个基于生成对抗网络（GANs）框架的符号多轨音乐生成模型。这三个模型在基本假设和相应的网络架构上存在差异，分别称为 jamming 模型、composer 模型和 hybrid 模型。我们使用超过十万小节的摇滚音乐数据集对所提出的模型进行了训练，并将其用于生成五个轨道的钢琴卷帘：低音、鼓、吉他、钢琴和弦乐。除了主观用户研究外，我们还提出了一些轨道内和轨道间客观度量指标来评估生成结果。我们展示了我们的模型可以生成四小节连贯的音乐（即没有人类输入）。我们还将我们的模型扩展到人工智能协作音乐生成：给定一个由人类创作的特定轨道，我们可以生成四个附加轨道来伴奏。所有代码、数据集和渲染音频样本均可在 https://salu133445.github.io/musegan/ 上获得。

简介

在人工智能领域，生成逼真和美学的作品被视为最令人兴奋的任务之一。近年来，在生成图像、视频和文本方面取得了重大进展，特别是使用生成对抗网络（GANs）（Goodfellow 等人，2014 年；Radford、Metz 和 Chintala，2016 年；Vondrick、Pirsiavash 和 Torralba，2016 年；Saito、Matsumoto 和 Saito，2017 年；Yu 等人，2017 年）。类似的尝试也已经用于生成符号音乐，但由于以下原因，这项任务仍然具有挑战性。

首先，音乐是一门时间艺术。如图 1 所示，音乐具有分层结构，较高层次的构建块（例如乐句）由较小的循环模式（例如小节）组成。人们在听音乐时会关注与连贯性、节奏、紧张感和情感流相关的结构模式（Herremans 和 Chew，2017）。因此，考虑时间结构的机制至关重要。

第二，音乐通常由多种乐器/轨道组成。一个现代管弦乐队通常包括四个不同的部分：

铜管、弦乐、木管和打击乐器；一支摇滚乐队通常包括一个贝斯、一个鼓组、吉他和可能还有一个歌唱部分。这些轨道相互密切地交互作用，并相互依存地随着时间展开。在音乐理论中，我们还可以找到大量关于音响关系的组合讨论，例如和声和对位法。

最后，音符通常被分组成和弦、琶音或旋律，因此引入音符的时间顺序并不自然适合于复调音乐。因此，自然语言生成和单声部音乐生成的成功不一定适用于复调音乐生成。

因此，在早期的相关工作（请参见相关工作部分的简要调查）中，大多数选择通过某些方式简化符号音乐生成，以使问题易于处理。这些简化包括：仅生成单轨单声部音乐，为复调音乐引入音符的时间顺序，将多声部音乐生成为几个单声部旋律的组合等。

我们的目标是尽可能避免这种简化。本质上，我们的目标是生成具有 1）和谐和节奏结构、2）多轨互相依赖和 3）时间结构的多声部复调音乐。

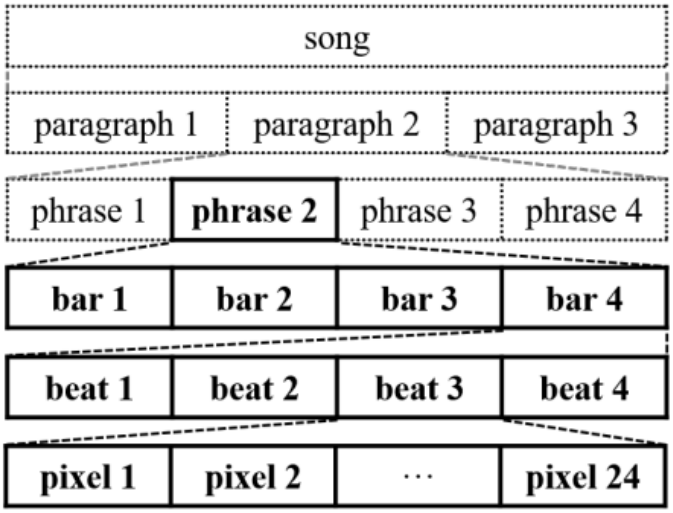


图 1：一首音乐作品的层次结构。

为了结合时间模型，我们提出了两种方法来应对不同的场景：一种是生成全新的音乐（即没有人类输入），另一种是在给定先验人类轨道下学习跟随其潜在的时间结构。为了处理轨道之间的相互作用，我们提出了三种方法，基于我们对流行音乐如何组成的理解：一种通过它们各自的私有生成器（每个轨道一个）独立生成轨道；另一种通过一个生成器联合生成所有轨道；另一种通过每个轨道的私有生成器生成每个轨道，并在轨道之间添加共享输入，以期引导轨道共同协调和和谐。为了应对音符的分组，我们将小节而不是音符视为基本构成单元，并使用转置卷积神经网络（CNNs）逐个生成音乐小节，这种方法被认为可以很好地寻找局部、平移不变的模式。

我们还提出了一些轨道内部和轨道间的客观度量方法，并使用它们来监控学习过程并量化地评估不同提出的模型生成的结果。我们还报告了一个涉及 144 名听众的用户研究，

用于主观评估结果。

我们将我们的模型称为多轨序列生成对抗网络，简称 MuseGAN。虽然本文关注音乐生成，但设计相当通用，我们希望它也能被应用于其他领域的多轨序列生成。

我们的贡献如下：

- 我们提出了一种基于 GAN 的新型模型，用于多轨序列生成。
- 我们将所提出的模型应用于生成符号音乐，据我们所知，这是第一个可以生成多轨复调音乐的模型。
- 我们将所提出的模型扩展到了轨道条件生成，可以应用于人工智能协作音乐生成或音乐伴奏。
- 我们提供了 Lakh Pianoroll 数据集（LPD），其中包含 173,997 个独特的多轨钢琴卷帘，来自 Lakh Midi 数据集（LMD）（Raffel 2016）。
- 我们提出了一些轨道内部和轨道间的客观度量指标，用于评估人工符号音乐。所有代码、数据集和生成的音频样本均可在我们的项目网站上找到。

生成式对抗网络

GAN 的核心概念是通过构建两个网络（生成器和判别器）实现对抗学习（Goodfellow 等人，2014）。生成器将从先验分布中采样的随机噪声 z 映射到数据空间。判别器被训练来区分真实数据和由生成器生成的数据，而生成器则被训练来欺骗判别器。训练过程可以形式化地建模为生成器 G 和判别器 D 之间的两个玩家的极小极大博弈：

$$\min_G \max_D \mathbf{E}_{\mathbf{x}\sim p_d}[\log(D(\mathbf{x}))] + \mathbf{E}_{\mathbf{z}\sim p_z}[1-\log(D(G(\mathbf{z})))], \quad (1)$$

其中 pa 和 p 分别表示真实数据的分布和 z 的先验分布。

在后续的研究中（Arjovsky，Chintala 和 Bottou2017），他们认为使用 Wasserstein 距离或 Earth Movers 距离来代替原始公式中使用的 Jensen-Shannon 散度可以稳定训练过程并避免模式崩溃。为了强制实施 K-Lipschitz 约束，WassersteinGAN 使用权重剪切，但后来发现它会导致优化困难。然后在（Gulrajani 等，2017）中提出了一个额外的梯度惩罚项，用于判别器的目标函数。D 的目标函数变为：

$$\mathbf{E}_{\mathbf{x}\sim p_d}[D(\mathbf{x})] - \mathbf{E}_{\mathbf{z}\sim p_z}[D(G(\mathbf{z}))] + \mathbf{E}_{\hat{\mathbf{x}}\sim p_{\hat{\mathbf{x}}}}[(\nabla_{\hat{\mathbf{x}}}\|\hat{\mathbf{x}}\| - 1)^2], \quad (2)$$

其中 p 是在从 pa 和 pg（模型分布）中采样的点对之间均匀采样的直线上定义的。结果

WGAN-GP 模型被发现收敛更快，优化效果更好，需要更少的参数调整。因此，在这项工作中，我们采用 WGAN-GP 模型作为我们的生成模型。

提出的模型

沿用（Yang，Chou 和 Yang，2017）的方法，我们将小节视为基本的构成单元，因为在小节的边界处通常会出现和声变化（例如，和弦变化），而且人类在创作歌曲时经常使用小节作为构建块。

数据表示法

为了建模多轨复调音乐，我们建议使用多轨钢琴卷帘表示。如图 2 所示，钢琴卷帘表示是一个二进制值的评分表状矩阵，表示不同时间步骤上音符的存在情况，多轨钢琴卷帘被定义为一组不同轨道的钢琴卷帘。

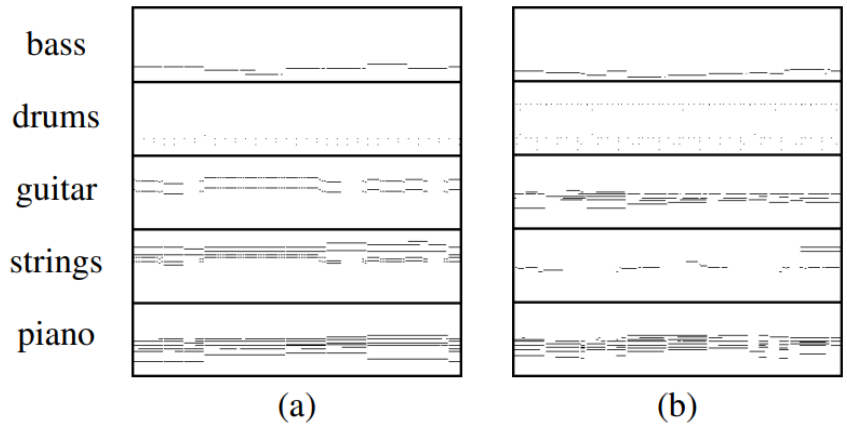


图 2：包含五个轨道的两个四小节音乐片段的多轨钢琴卷帘表示。水平轴表示时间，垂直轴表示音符（从低音到高音）。黑色像素表示在该时间步骤播放了特定的音符。

形式上，一个小节的 M 轨钢琴卷帘被表示为张量$x \in \{0,1\}^{RxSxM}$，其中 R 和 S 分别表示小节中的时间步骤数和音符候选数。T 个小节的 M 轨钢琴卷帘被表示为$\vec{x} = \{\vec{x}^{(t)}\}_{t=1}^{T}$，其中$\vec{x}^{(t)} \in \{0,1\}^{RxSxM}$表示第 t 个小节的多轨钢琴卷帘。

需要注意的是，每个小节、每个轨道的钢琴卷帘，对于真实数据和生成数据，都表示为一个固定大小的矩阵，这使得使用卷积神经网络成为可能。

建立多轨制相互依存关系的模型

在我们的经验中，有两种常见的创作音乐的方式。给定一组演奏不同乐器的音乐家，他们可以通过即兴演奏音乐来创建音乐，没有预先定义的编排，也就是所谓的即兴创作。或者，我们可以有一位作曲家根据和声结构和乐器的知识来编排乐器。音乐家将按照作曲的要求演奏音乐。我们设计了三种模型，对应于这些创作方式。

即兴创作模型

多个生成器独立地工作，从私有随机向量 zi，i = 1, 2, ...，M 中生成其自己轨道的音乐，其中 M 表示生成器（或轨道）的数量。这些生成器从不同的判别器接收评价（即反向传播的监督信号）。如图 3(a)所示，为了生成 M 轨道的音乐，我们需要 M 个生成器和 M 个判别器。

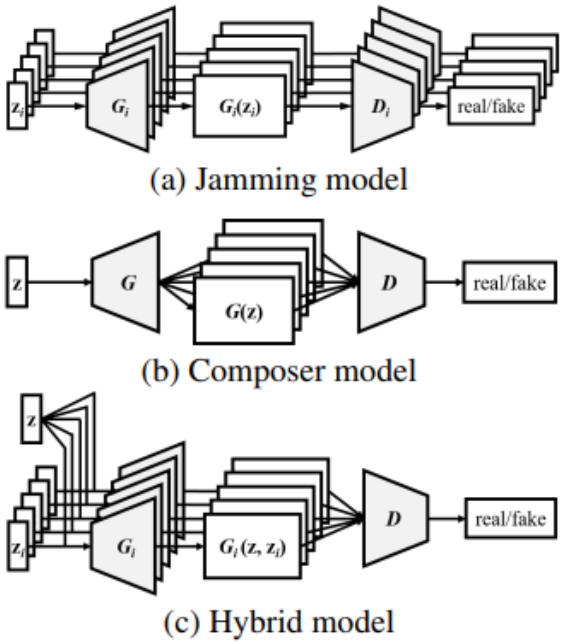


图 3：用于生成多轨数据的三个 GAN 模型。请注意，我们没有显示将馈送给鉴别器的真实数据 x。

作曲家模型

一个单独的生成器创建一个多通道钢琴卷帘，每个通道表示一个特定的轨道，如图 3(b)所示。这种模型只需要一个共享的随机向量 z（可以看作是作曲家的意图）和一个判别器，它集体检查 M 轨道以确定输入音乐是真实的还是假的。无论 M 的值是多少，我们总是只需要一个生成器和一个判别器。

混合模型

结合即兴演奏和作曲的思想，我们进一步提出了混合模型。如图 3(c)所示，M 个生成器中的每一个都以跨轨随机向量 z 和内轨随机向量 z 作为输入。我们希望跨轨随机向量可以协调不同音乐家的生成，即G_i，就像作曲家一样。此外，我们只使用一个判别器来集体评估 M 轨道。也就是说，我们需要 M 个生成器和一个判别器。

作曲家模型和混合模型之间的一个主要区别在于灵活性——在混合模型中，我们可以

使用不同的网络架构（例如，层数、过滤器大小）和不同的输入来生成 M 个生成器。因此，我们可以例如改变一个特定轨道的生成，而不会失去跨轨依赖性。

建立时间结构的模型

上述模型只能逐小节地生成多轨音乐，不同小节之间可能没有连贯性。我们需要一个时间模型来生成几小节长的音乐，例如一个乐句（参见图 1）。我们设计了两种方法来实现这一点，如下所述。

从头生成

第一种方法旨在通过将小节进展视为生成器的另一个维度来生成固定长度的乐句。生成器由两个子网络组成，即时间结构生成器 Gtemp 和小节生成器 Gbar，如图 4(a)所示。Gtemp 将一个噪声向量 z 映射到一些潜在向量的序列$\vec{z} = \{\vec{z}^{(t)}\}_{t=1}^{T}$。生成的$\vec{z}$应该携带时间信息，然后被 Gbar 依次用于生成钢琴卷帘（即逐小节生成）：

$$G(\mathbf{z}) = \left\{G_{\text{bar}}\left(G_{\text{temp}}(\mathbf{z})^{(t)}\right)\right\}_{t=1}^{T}. \qquad (3)$$

我们注意到，类似的想法已经被 Saito、Matsumoto 和 Saito (2017)用于视频生成。

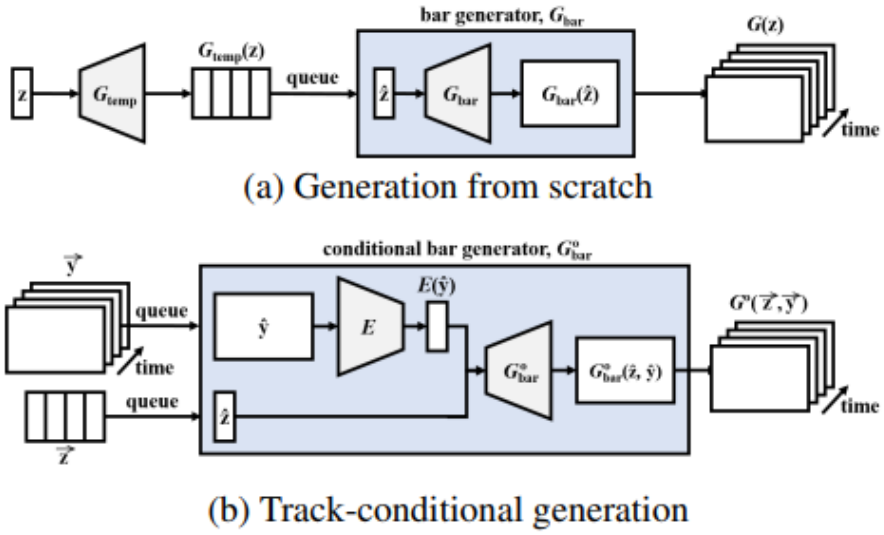


图 4：我们工作中采用的两种时间模型。请注意，仅显示了生成器。

轨道条件生成

第二种方法假设给定了一个特定轨道的小节序列$\vec{y}$，并尝试学习该轨道之下的时间结构并生成其余轨道（以及完成歌曲）。如图 4(b)所示，轨道条件生成器 Go 使用条件小节生成器 Gbar 逐个生成小节。然后，Gbar 生成其余小节的多轨钢琴卷帘，Gbar 采用两个输入，即条件$\vec{y}^{(t)}$和一个随时间变化的随机噪声$\vec{z}^{(t)}$。

为了实现带有高维条件的条件生成，需要训练额外的编码器 E，将 y(t)映射到 z(t)的空间中。值得注意的是，类似的方法已经被 Yang、Chou 和 Yang (2017)采用。整个过程可以表示为：

$$G^{\circ}\left(\overrightarrow{\mathbf{z}}, \overrightarrow{\mathbf{y}}\right)=\left\{G_{\mathrm{bar}}^{\circ}\left(\overrightarrow{\mathbf{z}}^{(t)}, E\left(\overrightarrow{\mathbf{y}}^{(t)}\right)\right)\right\}_{t=1}^{T} . \quad (4)$$

请注意，编码器预计从给定的轨道中提取跨轨道特征而不是内部轨道特征，因为内部轨道特征不应该对生成其他轨道有用。

据我们所知，以这种方式结合时间模型是新的。它可以应用于人工智能与人类合作生成，或音乐伴奏。

MuseGAN

我们现在介绍 MuseGAN，这是所提出的多轨和时间模型的整合和扩展。如图 5 所示，MuseGAN 的输入，表示为 z，由四个部分组成：跨轨时间独立随机向量 z、内轨时间独立随机向量 Zi、跨轨时间相关随机向量 Zt 和内轨时间相关随机向量 Zi,t。

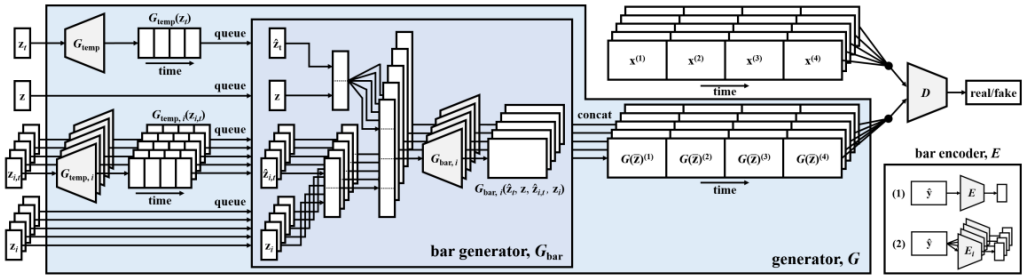


图 5：所提出的 MuseGAN 模型多轨顺序数据生成的系统图。

对于轨道 i（i = 1，2，...，M），共享的时间结构生成器 Gtemp 和私有的时间结构生成器 Gtemp，i 分别将时间相关随机向量 Zt 和 Zi，t 作为它们的输入，并且每个输出包含跨轨时间信息和内轨时间信息的潜在向量序列。输出序列（潜在向量），连同时间独立随机向量 z 和 Zi，被连接在一起并馈送到小节生成器 Gbar，然后逐个生成钢琴卷帘。生成过程可以表示为：

$$G(\bar{\mathbf{z}})=\left\{G_{\mathrm{bar},i}\left(\mathbf{z}, G_{\mathrm{temp}}\left(\mathbf{z}_t\right)^{(t)}, \mathbf{z}_i, G_{\mathrm{temp},i}\left(\mathbf{z}_{i,t}\right)^{(t)}\right)\right\}_{i,t=1}^{M,T} . \quad (5)$$

对于轨道条件场景，额外的编码器 E 负责从用户提供的轨道中提取有用的跨轨道特征。这可以类比进行，由于空间限制，我们省略了详细信息。

实现

数据集

我们在这项工作中使用的钢琴卷帘数据集是从 Lakh MIDI 数据集（LMD）（Raffel 2016）中导出的，这是一个包含 176,581 个唯一 MIDI 文件的大型集合。我们将 MIDI 文件转换为多轨钢琴卷帘。对于每个小节，我们将高度设置为 128，将宽度（时间分辨率）设置为 96，以建模常见的时间模式，例如三连音和 16 分音符。我们使用 Python 库 pretty midi（Raffel 和 Ellis 2014）解析和处理 MIDI 文件。我们将结果数据集命名为 Lakh Pianoroll Dataset（LPD）。我们还提供了子集 LPD-matched，它源自于 LMD-matched，这是 45,129 个 MIDI 的子集，与 Million Song Dataset（MSD）（Bertin-Mahieux 等人 2011）中的条目匹配。这两个数据集以及元数据和转换工具都可以在项目网站上找到。

数据预处理

由于这些 MIDI 文件是从网络上爬取的，大部分是用户生成的（Raffel 和 Ellis 2016），因此数据集相当嘈杂。因此，在这项工作中，我们使用 LPD-matched，并执行三个步骤进行进一步的清理（见图 6）。

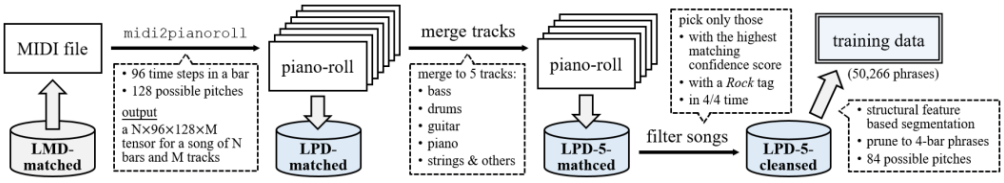


图 6：数据集准备和数据预处理过程的示意图。

首先，一些轨道倾向于在整首歌曲中仅播放几个音符。这增加了数据稀疏性并阻碍了学习过程。我们通过合并类似乐器的轨道（通过对它们的钢琴卷帘求和）来处理这种数据不平衡问题。每个多轨钢琴卷帘都被压缩成五个轨道：低音、鼓、吉他、钢琴和弦乐器。这样做会给我们的数据引入噪声，但是经验上我们发现这比有空小节要好。完成此步骤后，我们得到了 LPD-5-matched，其中包含 30,887 个多轨钢琴卷帘。

由于没有明确的方法来确定哪个轨道演奏旋律，哪个轨道演奏伴奏（Raffel 和 Ellis 2016），因此我们不能像一些之前的研究（Chu、Ur-tasun 和 Fidler 2017；Yang、Chou 和 Yang 2017）那样将轨道分类为旋律、节奏和鼓轨道。

其次，我们利用 LMD 和 MSD 中提供的元数据，仅选择具有更高匹配置信度、为摇滚歌曲和 4/4 拍子的钢琴卷帘。完成此步骤后，我们得到了 LPD-5-cleansed。

最后，为了获得具有音乐意义的短语以训练我们的时间模型，我们使用现代算法结构特征（Serrà 等人 2012）对钢琴卷帘进行分段，并相应地获得短语。在这项工作中，我们将四个小节视为一个短语，并将更长的片段修剪成适当的大小。我们总共获得了 50,266 个短语的训练数据。值得注意的是，尽管我们的模型仅用于生成固定长度的片段，但轨道条

件模型能够根据输入生成任意长度的音乐。

由于非常低和非常高的音符不常见，我们将低于 C1 或高于 C8 的音符丢弃。因此，目标输出张量（即一个片段的人工钢琴卷帘）的大小为 4（小节）×96（时间步骤）×84（音符）×5（轨道）。（有关训练数据中样本钢琴卷帘，请参见附录 A。）

模型设置

G 和 D 都是基于深度卷积神经网络实现的。G 首先在时间轴上增加，然后在音高轴上增加，而 D 则相反地进行压缩。根据（Gulrajani 等人 2017）的建议，我们每更新 5 次 D 后更新 G，仅对 G 应用批量归一化。每个生成器的输入随机向量的总长度固定为 128.9。每个模型的训练时间小于 24 小时，使用 Tesla K40m GPU。在测试阶段，我们通过零阈值将 G 的输出进行二值化，该输出在最后一层使用 tanh 作为激活函数。（有关更多详细信息，请参见附录 B。）

评估的客观指标

为了评估我们的模型，我们设计了几个指标，可以针对真实数据和生成数据计算，包括四个轨道内的指标和一个轨道间的指标（最后一个）：

- EB：空小节比率（以%计算）。
- UPC：每小节使用的音高类别数量（从 0 到 12）。
- QN：合格音符比率（以%计算）。我们认为不短于三个时间步长（即 32 分音符）的音符是合格音符。QN 显示音乐是否过于分散。
- DP，或称鼓点模式：在 4/4 拍子的摇滚歌曲中常见的 8 或 16 节奏模式中的音符比率（以%计算）。
- TD：或称为音程距离（Harte、Sandler 和 Gasser 2006）。它衡量一对轨道之间的谐和关系。较大的 TD 意味着轨道间的和声关系较弱。

通过比较从真实数据和生成数据计算得出的值，我们可以了解生成器的性能。这个概念类似于 GAN 中的概念——随着训练过程的进行，真实数据和生成数据的分布（因此统计学）应该变得更加接近。

训练数据的分析

我们将这些指标应用于训练数据，以更好地了解我们的训练数据。结果如表 1 和表 2 的第一行所示。EB 的值表明将轨道分类为五个家族是合适的。从 UPC 中，我们发现低音倾向于演奏旋律，导致 UPC 低于 2.0，而吉他、钢琴和弦乐器倾向于演奏和弦，导致 UPC 高于 3.0。较高的 QN 值表明转换后的钢琴卷帘不过于分散。从 DP 中，我们看到超过 88％

的鼓点音符处于 8 或 16 节奏模式中。当测量旋律类似轨道（主要是低音）和和弦类似轨道（主要是吉他、钢琴或弦乐器之一）之间的距离时，TD 的值约为 1.50，而对于两个和弦类似轨道，则约为 1.00。值得注意的是，如果我们通过随机配对两个特定轨道的小节来洗牌训练数据，TD 会略微增加，这表明 TD 确实捕捉了轨道间的和声关系。

		empty bars (**EB**; %)					used pitch classes (**UPC**)				qualified notes (**QN**; %)				**DP** (%)
		B	**D**	**G**	**P**	**S**	**B**	**G**	**P**	**S**	**B**	**G**	**P**	**S**	**D**
training data		8.06	8.06	19.4	24.8	10.1	1.71	3.08	3.28	3.38	90.0	81.9	88.4	89.6	88.6
from scratch	jamming	6.59	2.33	18.3	22.6	6.10	1.53	3.69	4.13	4.09	71.5	56.6	62.2	63.1	93.2
	composer	0.01	28.9	1.34	0.02	0.01	2.51	4.20	4.89	5.19	49.5	47.4	49.9	52.5	75.3
	hybrid	2.14	29.7	11.7	17.8	6.04	2.35	4.76	5.45	5.24	44.6	43.2	45.5	52.0	71.3
	ablated	92.4	100	12.5	0.68	0.00	1.00	2.88	2.32	4.72	0.00	22.8	31.1	26.2	0.0
track-conditional	jamming	4.60	3.47	13.3	—	3.44	2.05	3.79	—	4.23	73.9	58.8	—	62.3	91.6
	composer	0.65	20.7	1.97	—	1.49	2.51	4.57	—	5.10	53.5	48.4	—	59.0	84.5
	hybrid	2.09	4.53	10.3	—	4.05	2.86	4.43	—	4.32	43.3	55.6	—	67.1	71.8

表 1：轨内评价（B：贝斯，D：鼓，G：吉他，P：钢琴，S：弦乐；靠近第一行的数值更好。）

		tonal distance (**TD**)					
		B-G	**B-S**	**B-P**	**G-S**	**G-P**	**S-P**
train.		1.57	1.58	1.51	1.10	1.02	1.04
train. (shuffled)		1.59	1.59	1.56	1.14	1.12	1.13
from scratch	jam.	1.56	1.60	1.54	1.05	0.99	1.05
	comp.	1.37	1.36	**1.30**	0.95	0.98	0.91
	hybrid	**1.34**	**1.35**	1.32	**0.85**	**0.85**	**0.83**
track-condi-tional	jam.	1.51	1.53	1.50	1.04	0.95	1.00
	comp.	1.41	**1.36**	1.40	**0.96**	1.01	**0.95**
	hybrid	**1.39**	**1.36**	**1.38**	**0.96**	**0.94**	**0.95**

表 2：轨道间评估（数值越小越好）

实验和结果

结果示例

图 7 展示了由作曲家和混合模型生成的六个短语的钢琴卷帘。（请参见附录 C 以获取更多的钢琴卷帘样本。）我们的项目网站上可以找到一些渲染的音频样本。

一些观察结果：

- 轨道通常都在同一个音乐音阶中演奏。
- 在一些样本中可以观察到类似和弦的音程。

- 低音通常演奏最低音符，大部分时间是单声部的（即演奏旋律）。
- 鼓通常具有 8 或 16 节奏模式。
- 吉他、钢琴和弦乐器倾向于演奏和弦，它们的音高有时会重叠（创建黑色线条），这表明它们之间存在良好的和声关系。

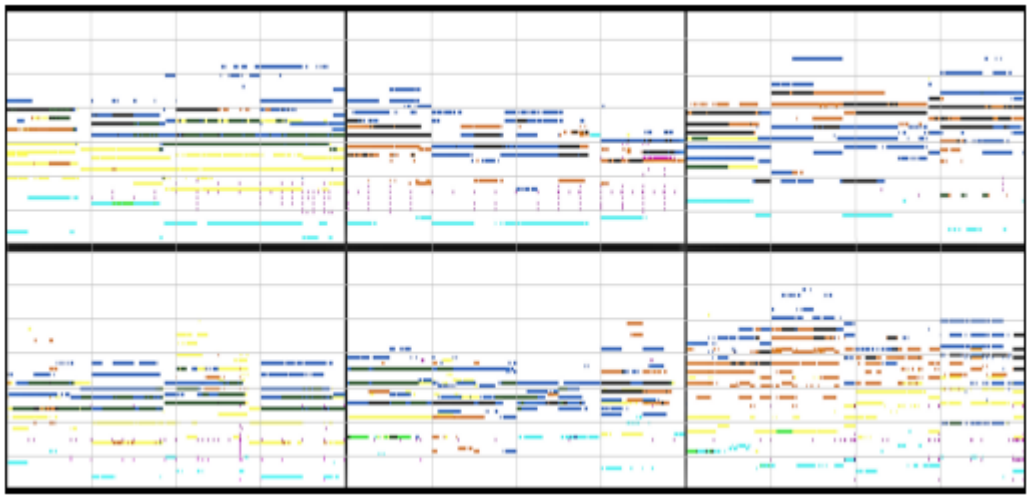


图 7：作曲家模型（上排）和混合模型（下排）生成的示例结果，均从头开始生成（最佳观看效果为彩色——青色：低音，粉色：鼓，黄色：吉他，蓝色：弦乐器，橙色：钢琴）

客观评价

为了检查我们的模型，我们使用每个模型生成了 20,000 个小节，并根据提出的客观指标进行评估。结果如表 1 和表 2 所示。请注意，对于条件生成情况，我们使用钢琴轨道作为条件，生成其他四个轨道。为了比较，我们还包括了一个去除批量归一化层的作曲家模型的消融版本的结果。这个去除了批量归一化层的模型几乎没有学到任何东西，因此其值可以作为参考。

对于轨道内的指标，我们发现即兴模型倾向于表现最好。这可能是因为即兴模型中的每个生成器都被设计为只关注自己的轨道。除了消融模型之外，所有模型在 DP 方面表现良好，这表明鼓确实捕捉到了训练数据中的一些节奏模式，尽管作曲家模型和混合模型中鼓的 EB 相对较高。从 UPC 和 QN 中，我们可以看到所有模型都倾向于使用更多的音高类别，并且产生的合格音符比训练数据少。这表明可能产生了一些噪声，并且生成的音乐包含大量过于分散的音符，这可能是由于我们将 G 的连续值输出二值化（创建二值钢琴卷帘）的方式所致。我们目前没有解决方案，将其作为未来的工作。

对于轨道间的指标 TD（表 2），我们可以看到作曲家模型和混合模型的值相对较低，而即兴模型的值相对较高。这表明即兴模型生成的音乐轨道之间的和声关系较弱，而作曲

家模型和混合模型在跨轨道和声关系方面可能更为适合多轨生成。此外，我们可以看到作曲家模型和混合模型在不同的轨道组合下表现相似。这是令人鼓舞的，因为我们知道混合模型可能没有为其灵活性而牺牲性能。

训练过程

为了获得训练过程的洞见，我们首先研究了作曲家模型从头开始生成的情况（其他模型有类似的行为）。图 9（a）显示了 D 的训练损失作为训练步骤的函数。我们可以看到，它在开始时迅速下降，然后达到饱和。然而，在图中标记为 B 的点之后，有一个轻微的增长趋势，表明 G 在那之后开始学习一些东西。

我们在图 8 中展示了在图 9（a）上标记的五个点处生成的钢琴卷帘，从中我们可以观察到随着训练过程的展开，生成的钢琴卷帘如何发展。例如，我们可以看到 G 相当早地掌握了每个轨道的音高范围，并且在 B 点开始产生一些音符（虽然分散但在适当的音高范围内），而不是在 A 点产生噪声。在 B 点，我们已经可以看到在低音部分（具有较低音高）聚集的点簇。在 C 点之后，我们可以看到吉他、钢琴和弦乐器开始学习音符的持续时间，并开始产生较长的音符。这些结果表明，随着训练过程的进行，G 确实变得更好了。

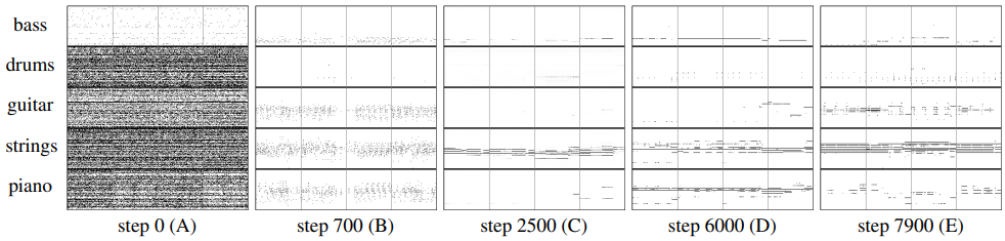


图 8：作曲家模型从头开始生成时，生成的钢琴卷帘随着更新步骤的变化而演变的情况。

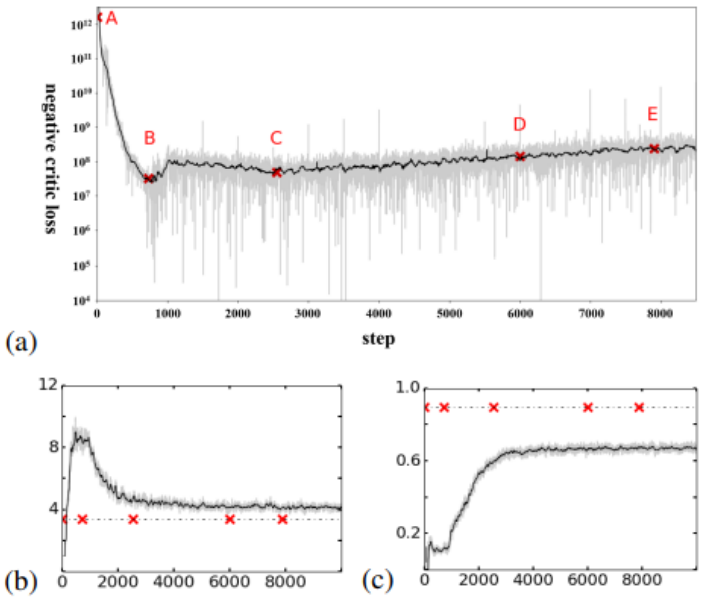


图 9：（a）判别器的训练损失，（b）弦乐器轨道的 UPC 和（c）QN，针对作曲家模型从头开始生成的情况。灰色和黑色曲线分别是原始值和平滑值（通过中值滤波器平滑）。在（b）和（c）中的虚线表示从训练数据计算出的值。

我们还在图 9 中展示了两个客观指标沿着训练过程的变化。从（b）中，我们可以看到 G 最终可以学习到适当数量的音高类别；从（c）中，我们可以看到 QN 保持相对较低，低于训练数据，这表明我们的 G 还有改进的空间。这些结果表明，研究人员可以使用这些指标在进行主观测试之前研究生成的结果。

用户研究

最后，我们进行了一个听力测试，招募了 144 名受试者通过我们的社交圈从互联网上招募。其中 44 人根据一份简单的调查问卷被认定为“专业用户”，该问卷调查了他们的音乐背景。每个受试者必须以随机顺序听取九个音乐片段。每个片段由所提出模型之一生成的三个四小节乐句构成，并由十六分音符量化。受试者使用五点 Likert 量表对片段进行评分，评价它们是否具有 1）愉悦的和声，2）统一的节奏，3）清晰的音乐结构，4）连贯性以及 5）总体评价。

从表 3 中显示的结果可以看出，专业用户和非专业用户都更喜欢混合模型进行从头开始的生成，而专业用户更喜欢混合模型进行条件生成，而非专业用户更喜欢即兴模型进行条件生成。此外，对于从头开始的生成，作曲家模型和混合模型在和谐标准上得分更高，而即兴模型得分较低，这表明作曲家模型和混合模型在处理轨道间的相互依赖方面表现更好。

			H	R	MS	C	OR
from scratch	non-pro	jam.	2.83	3.29	2.88	2.84	2.88
		comp.	3.12	**3.36**	2.95	3.13	3.12
		hybrid	**3.15**	3.33	**3.09**	**3.30**	**3.16**
	pro	jam.	2.31	3.05	2.48	2.49	2.42
		comp.	2.66	3.13	2.68	2.63	2.73
		hybrid	**2.92**	**3.25**	**2.81**	**3.00**	**2.93**
track-conditional	non-pro	jam.	**2.89**	**3.44**	2.97	**3.01**	**3.06**
		comp.	2.70	3.29	**2.98**	2.97	2.86
		hybrid	2.78	3.34	2.93	2.98	3.01
	pro	jam.	2.44	**3.32**	2.67	2.72	2.69
		comp.	2.35	3.21	2.59	2.67	2.62
		hybrid	**2.49**	3.29	**2.71**	**2.73**	**2.70**

表 3：用户研究结果（H：和谐，R：节奏，MS：音乐结构，C：连贯，OR：总体评分）

从表 3 中显示的结果可以看出，专业用户和非专业用户都更喜欢混合模型进行从头开始的生成，而专业用户更喜欢混合模型进行条件生成，而非专业用户更喜欢即兴模型进行条件生成。此外，对于从头开始的生成，作曲家模型和混合模型在和谐标准上得分更高，而即兴模型得分较低，这表明作曲家模型和混合模型在处理轨道间的相互依赖方面表现更好。

相关工作

利用生成对抗网络（GANs）进行视频生成

与音乐生成类似，视频生成也需要一个时间模型。我们的模型设计灵感来自于先前使用 GANs 进行视频生成的一些研究。VGAN（Vondrick、Pirsiavash 和 Torralba 2016）假设视频可以分解为动态前景和静态背景。他们使用 3D 和 2D CNN 分别在两个流中生成它们，并通过前景流生成的掩码组合结果。TGAN（Saito、Matsumoto 和 Saito 2017）使用一个时间生成器（使用卷积）生成一系列固定长度的潜在变量，然后逐一输入图像生成器，逐帧生成视频。MoCoGAN（Tulyakov 等人 2017）假设视频可以分解为内容（对象）和运动（对象的运动），并使用 RNN 来捕捉对象的运动。

符号音乐生成

最近，许多模型已被提出用于符号音乐生成，如（Briot，Hadjeres 和 Pachet 2017）所述。其中许多使用 RNN 生成不同格式的音乐，包括单声道旋律（Sturm 等人 2016）和四声部合唱曲（Hadjeres，Pachet 和 Nielsen 2017）。值得注意的是，RNN-RBM（Boulanger-Lewandowski，Bengio 和 Vincent 2012）是循环时间限制玻尔兹曼机（RTRBM）的一种泛化形式，能够生成单个轨道的多声部钢琴卷帘。Song from PI（Chu，Urtasun 和 Fidler 2017）使用分层 RNN 协调三个轨道，能够生成一个主旋律轨道和一个和弦标签轨道的伴奏谱。

一些最近的研究也开始探索使用 GANs 生成音乐。C-RNN-GAN（Mogren 2016）通过引入一些音符的排序，并在生成器和判别器中使用 RNN，生成一系列音符事件的多声部音乐。SeqGAN（Yu 等人 2017）将 GAN 和强化学习相结合，生成离散令牌序列。它已应用于生成单声道音乐，使用音符事件表示法。MidiNet（Yang，Chou 和 Yang 2017）使用条件卷积 GAN 生成旋律，根据预先给定的和弦序列，从头开始或以前几小节的旋律为条件。

总结

在这项工作中，我们提出了一种基于 GANs 框架的多轨序列生成的新型生成模型。我们还使用深度 CNN 实现了这样的模型，用于生成多声部钢琴卷帘。我们设计了几个客观度量标准，并展示了我们可以通过这些客观度量标准获得有关学习过程的见解。客观度量标准和主观用户研究表明，所提出的模型可以开始学习音乐的某些方面。尽管在音乐和美学上它可能仍然落后于人类音乐家的水平，但所提出的模型具有一些理想的特性，我们希

望后续的研究可以进一步改进它。

参考文献

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. arXiv preprint arXiv:1701.07875.

Bertin-Mahieux, T.; Ellis, D. P.; Whitman, B.; and Lamere, P. 2011. The Million Song Dataset. In ISMIR.

Boulanger-Lewandowski, N.; Bengio, Y.; and Vincent, P.2012. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In ICML.

Briot, J.-P.; Hadjeres, G.; and Pachet, F. 2017. Deep learning techniques for music generation: A survey. arXiv preprint arXiv:1709.01620.

Chu, H.; Urtasun, R.; and Fidler, S. 2017. Song from PI: A musically plausible network for pop music generation. In ICLR Workshop.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y.2014. Generative adversarial nets. In NIPS.

Gulrajani, I.; Ahmed, F; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of Wasserstein GANs. arXiv preprint arXiv:1704.00028.

Hadjeres, G.; Pachet, F.; and Nielsen, F. 2017. DeepBach: A steerable model for Bach chorales generation. In ICML.

Harte, C.; Sandler, M.; and Gasser, M. 2006. Detecting harmonic change in musical audio. In ACM MM workshop on Audio and music computing multimedia.

Herremans, D., and Chew, E. 2017. MorpheuS: generat-ing structured music with constrained patterns and tension.IEEE Trans. Affective Computing.

Mogren, O. 2016. C-RNN-GAN: Continuous recurrent neu-ral networks with adversarial training. In NIPS Worshop on Constructive Machine Learning Workshop.

Nieto, O., and Bello, J. P. 2016. Systematic exploration of computational music structure research. In ISMIR.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervisedrepresentation learning with deep convolutional generative adversarial networks. In ICLR.

Raffel, C., and Ellis, D. P. W. 2014. Intuitive analysis, cre-ation and manipulation of MIDI data with pretty midi. In ISMIR Late Breaking and Demo Papers.

Raffel, C., and Ellis, D. P. W. 2016. Extracting ground truth information from MIDI files: A MIDIfesto. In ISMIR.

Raffel, C. 2016. Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching. Ph.D. Dissertation, Columbia University.

Saito, M.; Matsumoto, E.; and Saito, S. 2017. Temporal generative adversarial nets with singular value clipping. In ICCV.

Serrà, J.; Mller, M.; Grosche, P.; and Arcos, J. L. 2012. Unsupervised detection of music boundaries by time series structure features. In AAAI.

Sturm, B. L.; Santos, J. F; Ben-Tal, O.; and Korshunova, I.2016. Music transcription modelling and composition using deep learning. In Conference on Computer Simulation of Musical Creativity.

Tulyakov, S.; Liu, M.; Yang, X.; and Kautz, J. 2017. MoCo-GAN: Decomposing motion and content for video genera-tion. arXiv preprint arXiv:1707.04993.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Gener-ating videos with scene dynamics. In NIPS.

Yang, L.-C.; Chou, S.-Y.; and Yang, Y.-H. 2017. MidiNet: Aconvolutional generative adversarial network for symbolic-domain music generation. In ISMIR.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient.In AAAI.

附录 A 训练数据的样本

我们在图 10 中展示了一些随机选择的训练数据中的钢琴卷积表示。

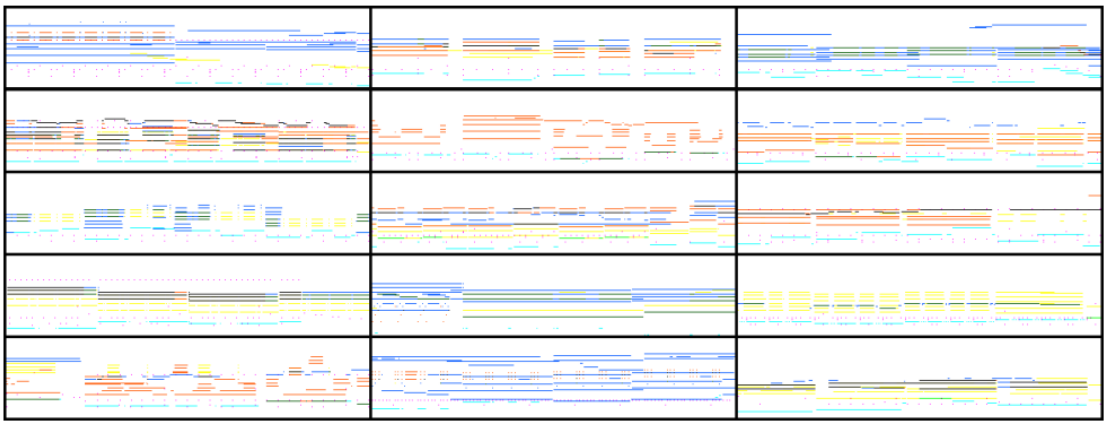


图 10：训练数据中的样本钢琴卷积表示（最好在彩色显示下观看：青色-低音，粉色-鼓，黄色-吉他，蓝色-弦乐器，橙色-钢琴）。

附录 B 实施细节

所提出模型的网络架构在表 4 中列出。以下是一些详细信息。

Input: $\mathbf{z} \in \Re^{32}$					
reshaped to (1)×32 channels					
transconv	1024	2	2	BN	ReLU
transconv	K_{temp}	3	1	BN	ReLU
Output: $G_{\text{temp}}(\mathbf{z}) \in \Re^{32\times K_{\text{temp}}}$ (K_{temp}-track latent vector)					

(a) the temporal generator G_{temp}

Input: $\mathbf{z} \in \Re^{128}$					
reshaped to (1, 1)×128 channels					
transconv	1024	2×1	(2, 1)	BN	ReLU
transconv	256	2×1	(2, 1)	BN	ReLU
transconv	256	2×1	(2, 1)	BN	ReLU
transconv	256	2×1	(2, 1)	BN	ReLU
transconv	128	3×1	(3, 1)	BN	ReLU
transconv	64	1×7	(1, 7)	BN	ReLU
transconv	K_{bar}	1×12	(1, 12)	BN	tanh
Output: $G_{\text{bar}}(\mathbf{z}) \in \Re^{96\times 84\times K_{\text{bar}}}$ (K_{bar}-track piano-roll)					

(b) the bar generator G_{bar}

Input: $\widetilde{\mathbf{x}} \in \Re^{4\times96\times84\times5}$ (real/fake piano-rolls of 5 tracks)				
reshaped to (4, 96, 84) × 5 channels				
conv	128	$2 \times 1 \times 1$	(1, 1, 1)	LReLU
conv	128	$3 \times 1 \times 1$	(1, 1, 1)	LReLU
conv	128	$1 \times 1 \times 12$	(1, 1, 12)	LReLU
conv	128	$1 \times 1 \times 7$	(1, 1, 7)	LReLU
conv	128	$1 \times 2 \times 1$	(1, 2, 1)	LReLU
conv	128	$1 \times 2 \times 1$	(1, 2, 1)	LReLU
conv	256	$1 \times 4 \times 1$	(1, 2, 1)	LReLU
conv	512	$1 \times 3 \times 1$	(1, 2, 1)	LReLU
fully-connected	1024			LReLU
fully-connected	1			
Output: $D(\widetilde{\mathbf{x}}) \in \Re$				

(c) the discriminator D

Input: $\mathbf{y} \in \Re^{96\times84}$ (piano-rolls of the given track)					
conv	16	1×12	(1, 12)	BN	LReLU
conv	16	1×7	(1, 7)	BN	LReLU
conv	16	3×1	(3, 1)	BN	LReLU
conv	16	2×1	(2, 1)	BN	LReLU
conv	16	2×1	(2, 1)	BN	LReLU
conv	16	2×1	(2, 1)	BN	LReLU
Output: $E(\mathbf{y}) \in \Re^{16}$					

(d) the encoder E

表 4：（a）时间生成器，（b）小节生成器，（c）鉴别器和（d）编码器的网络架构。对于卷积（conv）和转置卷积（transconv）层，值表示（从左到右）：过滤器数量，内核大小，步幅，批量归一化（BN）和激活函数。对于全连接层，值表示（从左到右）：隐藏节点数和激活函数。LReLU 代表泄漏的 ReLU。对于即兴演奏、作曲家和混合模型，（Ktemp，Kbar）分别为（1，1），（1，5），（5，1）。

随机向量

整个系统输入的随机向量总长度固定为 128，这可以是一个长度为 128 的向量，两个长度为 64 的向量或四个长度为 32 的向量，具体取决于所使用的模型。Gtemp 的输入随机向量与其输出潜向量具有相同的长度。因此，Gbar 的输入向量的总长度也为 128。

网络架构

生成器：Gtemp 由两个 1-D 转置卷积层组成，沿（小节间）时间轴。Gbar 由五个 1-D 转置卷积层沿（小节内）时间轴和两个 1-D 转置卷积层沿音高轴依次组成。在每个激活层之前添加了批量归一化（BN）层。

鉴别器：D 由五个 1-D 卷积层和一个全连接层组成。泄漏的修正线性单元（ReLU）的负斜率设置为 0.2。

编码器：E 具有与 G 相反的架构，并在相应的层上应用跳跃连接以加速训练过程。我们将每层的过滤器数量限制为 16，以压缩跨曲目相互依赖的表示。

训练

我们使用 Adam 优化器对整个网络进行端到端的训练，其中 a=0.001，β1=0.5，β2=0.9。根据（Gulrajani 等人，2017）的建议，我们在每更新 5 次 D 后更新 G（对于轨道条件生成模型也更新 E）。每个模型的训练时间不到 24 小时，使用 Tesla K40m GPU。

渲染音频

首先，我们将生成的钢琴卷积表示量化为 16 分音符，以避免音符过于碎片化。然后，我们将钢琴卷积表示转换为 MIDI 文件。然后在外部数字音频工作站中混合和渲染轨道，生成立体声音频文件。

附录 C 生成的钢琴卷积表示样本

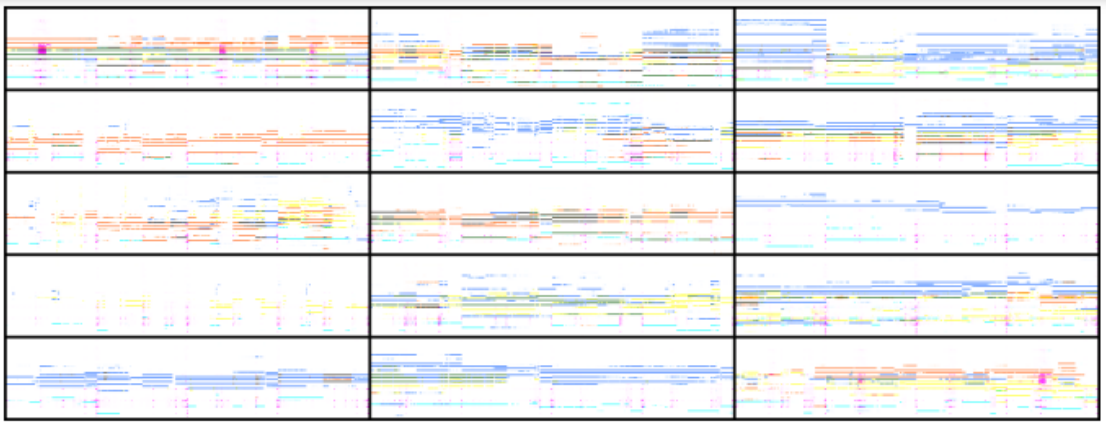
我们提供了我们模型生成的随机选择的钢琴卷积表示样本。

- 图 11 和图 12 分别显示由作曲家模型和混合模型从零开始生成的样本。
- 图 13 显示作曲家模型的轨道条件生成样本。请注意，我们在此处使用弦乐器轨道作为条件，而不是实验部分中使用的钢琴轨道，以展示我们模型的灵活性。

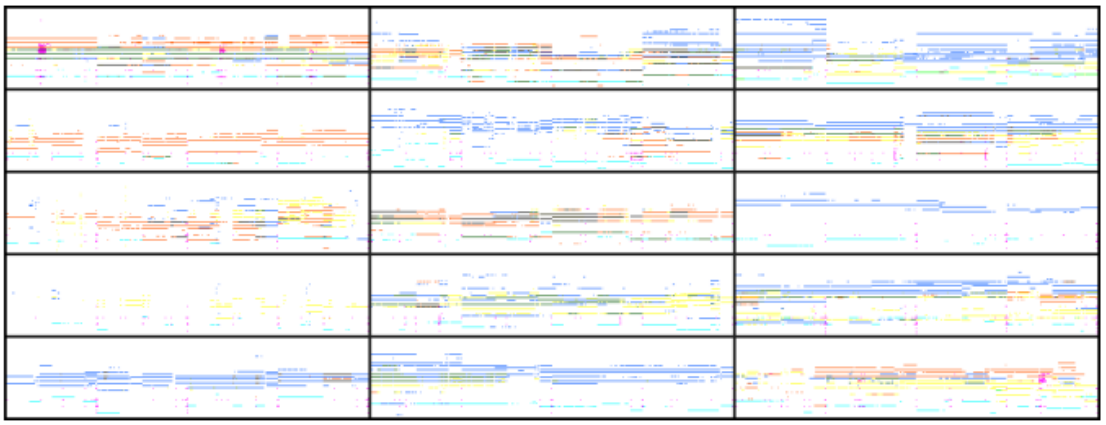
(a) original piano-rolls (before binarization)

(b) binarized piano-rolls

图 11：由作曲家模型从零开始生成的随机选择钢琴卷积表示（最好在彩色显示下观看：青色-低音，粉色-鼓，黄色-吉他，蓝色-弦乐器，橙色-钢琴）。在（b）中，我们通过零阈值对 G 的输出进行二值化，该模型在最后一层使用 tanh 作为激活函数。

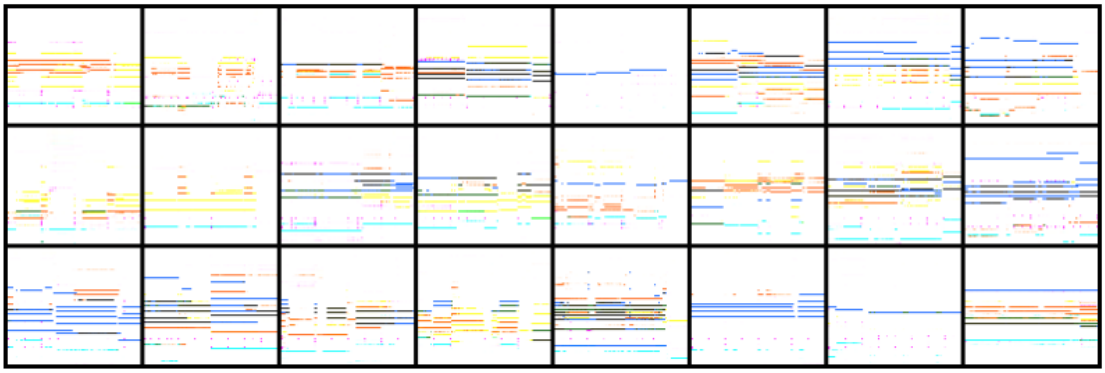


(a) original piano-rolls (before binarization)

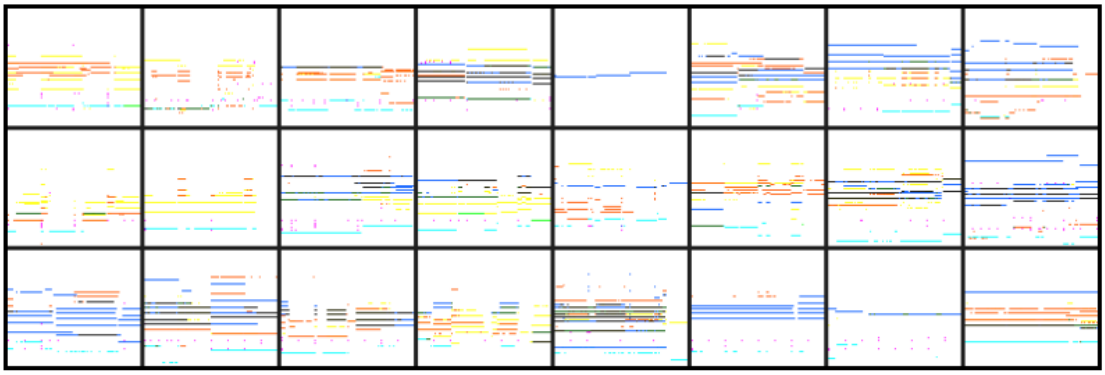


(b) binarized piano-rolls

图 12：由混合模型从零开始生成的随机选择钢琴卷积表示（最好在彩色显示下观看：青色-低音，粉色-鼓，黄色-吉他，蓝色-弦乐器，橙色-钢琴）。



(a) original piano-rolls (before binarization)



(b) binarized piano-rolls

图 13：在弦乐器轨道条件下，由作曲家模型生成的随机选择钢琴卷积表示（最好在彩色显示下观看：青色-低音，粉色-鼓，黄色-吉他，蓝色-弦乐器（条件），橙色-钢琴）。